



Métriques d'intégration pour le choix d'architectures dans la conception des systèmes mécatroniques

Aude Warniez

► To cite this version:

Aude Warniez. Métriques d'intégration pour le choix d'architectures dans la conception des systèmes mécatroniques. Automatique / Robotique. Ecole Centrale Paris, 2015. Français. NNT : 2015ECAP0030 . tel-01223145

HAL Id: tel-01223145

<https://theses.hal.science/tel-01223145>

Submitted on 2 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE
Présentée par WARNIEZ Aude

Pour l'obtention du
GRADE DE DOCTEUR

École Doctorale : École Centrale Paris (ED287)

Spécialité : **Sciences pour l'ingénieur**

Laboratoire d'accueil : **LISMMA (EA2336)**

**Métriques d'intégration pour le choix d'architectures
dans la conception des systèmes mécatroniques**

Soutenue le 12/05/2015

Devant un jury composé de

Mme. JANKOVIC Marija	Maître de Conférences HDR	Rapporteur
M. CHAPURLAT Vincent	Professeur des Universités	Rapporteur
M. LOTTIN Jacques	Professeur des Universités	Examineur
M. HUGEL Vincent	Professeur des universités	Examineur
M. SORIANO Thierry	Professeur des universités	Directeur de thèse
Mme. PENAS Olivia	Ingénieur de recherche	Co-encadrante de thèse

2014 - 2015

REMERCIEMENTS

Je tiens tout d'abord à remercier le directeur de cette thèse, M. Thierry SORIANO, pour m'avoir fait confiance malgré les connaissances plutôt légères, que j'avais au début de cette thèse sur les métriques, puis pour m'avoir guidée et conseillée tout au long de ces années.

Je remercie chaleureusement Jean-Yves CHOLEY, directeur du Laboratoire d'Ingénierie des Systèmes Mécaniques et MATériaux (LISMMA) pour m'avoir encouragée dans ce projet, en m'autorisant à réaliser cette thèse en parallèle de mes fonctions d'ingénieur d'études au sein de l'établissement, je le remercie aussi pour sa confiance indéfectible et le soutien qu'il a su m'apporter, dès que des difficultés sont survenues.

Un très grand merci à Olivia PENAS, qui m'a encadrée tout au long de cette thèse. Elle a toujours été disponible, à l'écoute, je la remercie pour ses qualités humaines, ses encouragements, sa confiance et son soutien. Ses conseils au quotidien, ses remarques et nos nombreuses discussions, ses nuits blanches pour les nombreuses relectures et corrections sont pour beaucoup dans le résultat final de ce travail. Pour tout cela merci...

Je tiens à remercier les différents membres de mon jury de thèse : Mme Marija JANKOVIC et M. Vincent CHAPURLAT qui m'ont fait l'honneur de juger de ce travail en tant que rapporteurs, M. Jacques LOTTIN et M. Vincent HUGEL d'avoir accepté d'être membres du jury. Leurs remarques et leurs questions ont permis d'améliorer ce travail et permettront assurément de le faire avancer.

Je remercie chaleureusement toute l'équipe mécatronique : Faïda MHENNI, Régis PLATEAUX, Moncef HAMMADI ainsi que Reda SELLAKH, Mireille MOINET et Stéphane DUGOWSON pour leur expertise, leur disponibilité, leurs conseils scientifiques qui m'ont permis d'avancer, ou enfin le temps qu'ils ont consacré à la relecture de ce manuscrit.

J'adresse toute ma gratitude et mes remerciements à Reda SELLAKH, Romain BARBEDIEENNE et Patrice LECLAIRE pour leur aide et leur collaboration, notamment pour la réalisation des extensions de profils, modèles et outils dans la partie expérimentale de cette thèse.

De plus, je souhaite remercier Peter HEHENBERGER et ACCM pour leur soutien et les échanges sur les travaux de recherche commun.

Je voudrais exprimer ma sincère reconnaissance envers les Pôles Techniques plus particulièrement à Mireille MOINET et au Pôle Instrumentation : Arkadiusz KOSECKI et Christophe BEN BRAHIM pour m'avoir permis de dégager du temps pour travailler sur cette thèse et pour la compréhension et le soutien dont ils ont su faire preuve durant ces années, afin que cette thèse se déroule dans les meilleures conditions.

Merci à tous les membres du laboratoire LISMMA qui ont participé de près ou de loin au bon déroulement de ces travaux et notamment Christel COMPAGNON pour sa bonne humeur et pour la gestion des aspects administratifs de ma thèse avec l'école doctorale.

Enfin, j'adresse tout particulièrement mes remerciements à mes amis et ma famille pour leur soutien, leur écoute et leur confiance. Ils sont toujours restés présents à mes côtés. Je remercie affectueusement mon compagnon Fabien pour son soutien, ses encouragements et sa patience notamment durant ces derniers mois lors de la rédaction de ce manuscrit.

Ces remerciements sont également dédiés à ma sœur Marie, avec qui j'ai pu partager, simultanément, les différentes émotions que l'on ressent lors de la rédaction d'une thèse.

Pour finir, un grand merci à mes parents qui grâce à leur amour, leur patience et à leur enseignement m'ont guidée jusqu'à présent.

Je dédie en particulier cette thèse à ma mère, qui me manque.

A ma mère,
A mon père,

Table des matières

Introduction	1
1 Problématique.....	2
2 Structure du mémoire	4
 Chapitre 1 - Contexte et problématique de l'évaluation en conception mécatronique.....	6
1 Introduction.....	7
1.1 La mécatronique.....	7
1.2 La conception des systèmes.....	8
1.2.1 Les principaux cycles de conception	9
1.2.1.1 Le cycle de conception en cascade.....	10
1.2.1.2 Le cycle de conception en V	10
1.2.1.3 Le cycle de conception en spirale.....	11
1.2.1.4 Le cycle de conception en Y.....	12
1.2.1.5 Le cycle de conception en X	12
1.2.2 Les méthodes de conception.....	13
1.2.2.1 Démarche séquentielle.....	13
1.2.2.2 Démarche itérative	14
1.2.2.3 Approches IS/ MBSE et langage SysML	14
1.2.3 Les différents points de vue de la modélisation durant la conception.....	17
1.2.3.1 Modélisation des niveaux de conception.....	17
1.2.3.2 Les modèles d'architectures.....	18
2 L'intégration dans la conception mécatronique	20
2.1 Intégration et complexité des systèmes mécatroniques.....	21
2.2 Spécificités de la conception mécatronique et méthode associée.....	23
2.3 Enjeux pour l'évaluation de la conception mécatronique	30
3 Aide aux choix de conception mécatronique : les métriques.....	30
3.1 Intérêt des métriques.....	30
3.2 Définition des métriques.....	32
4 Métriques dans le cycle de conception.....	34
4.1 Les métriques en phase de préconception	35
4.1.1 Les métriques de maturité	35

4.1.2	Les métriques de risques	36
4.1.3	Les métriques de coûts	37
4.1.4	Les métriques de planning.....	38
4.1.5	Les métriques d'efficacité (MoE)	39
4.1.5.1	Les Métriques (ou Mesures) de Performance (MoP).....	39
4.1.5.2	Les Métriques d'Usage/Conformité (MoS)	39
4.2	Les métriques dans la phase de conception préliminaire.....	40
4.2.1	Les métriques de modularité.....	40
4.2.2	Les métriques de fiabilité	41
4.2.3	Les métriques de complexité.....	43
	La complexité de conception	43
	La complexité (fonctionnelle) du produit	43
	La complexité mécatronique.....	44
	La complexité (structurale) de conception	44
	La complexité (taille, couplages et résolvabilité)	44
4.3	Les métriques dans la phase de conception détaillée	45
4.4	Les métriques en phase de vérification et validation	53
4.4.1	Les métriques de vérification	53
4.4.1.1	Les métriques de contrôle du processus.....	53
4.4.2	Les métriques de validation.....	54
4.4.2.1	Les Mesures de Performance Technique (TPM)	54
4.4.2.2	Les métriques d'usage « Suitability metrics »	55
4.4.2.3	Les métriques de satisfaction	55
4.4.2.4	Les métriques de traçabilité	55
4.4.2.5	Les métriques de validation expérimentation/calculs	56
5	Conclusions.....	56

Chapitre 2 - Métriques d'architectures pour l'intégration multi-domaine et multi-niveau 58

1	Introduction.....	59
1.1	Le contexte industriel.....	59
1.2	Problématique et plan.....	59
2	Etat de l'art	60
2.1	Définitions des architectures	60
2.2	Rôle et besoins de l'architecte système.....	61
2.3	Cycle de vie des architectures.....	63

2.3.1	Recherche préalable de concepts / Analyse fonctionnelle	63
2.3.2	Recherche et émergence d'architectures candidates.....	63
2.3.3	Formalisation/modélisation des architectures	64
2.3.4	Evaluation / Analyse des architectures	66
2.3.4.1	Cadre et définitions	67
2.3.4.2	Quality Function Deployment (QFD) – Maison de la qualité	67
2.3.4.3	DSM (Design Structure Matrix) – DMM (Domain Mapping Matrix)	70
2.3.4.4	Axiomatic design.....	72
2.3.4.5	Prise en compte des incertitudes.....	73
2.3.4.6	Evaluation des architectures logiques.....	74
2.3.4.7	Analyse de l'architecture physique (spatiale)	75
2.3.5	Aide à la décision pour le choix des architectures	75
2.3.5.1	Résolution d'un problème par satisfaction de contraintes.....	75
2.3.5.2	Aide à la décision multicritère.....	75
2.3.5.3	Méthodes basées sur l'intelligence artificielle.....	76
3	Métriques d'architectures développées	77
3.1	Métrique d'intégration des niveaux d'abstraction	78
3.1.1	Enjeux	78
3.1.2	Métrique d'intégration multi-niveau du système	80
3.1.3	Indicateur d'intégration des niveaux d'abstraction pour le système : IML_s ..	81
3.1.3.1	Définition	81
3.1.3.2	Expression.....	81
3.1.3.3	Bornes.....	81
3.1.3.4	Exemple	81
3.1.4	Indicateur d'intégration des niveaux d'abstraction par niveaux successifs : IML_n ..	82
3.1.4.1	Définition	82
3.1.4.2	Expression.....	83
3.1.4.3	Bornes.....	83
3.1.4.4	Exemple	83
3.1.5	Résultats, analyse et discussions.....	84
3.2	Métrique d'intégration multi-domaine.....	86
3.2.1	Enjeux	86
3.2.2	Métrique d'intégration multi-domaine du système.....	89
3.2.3	Indicateur d'intégration multi-domaine sur un niveau d'abstraction n : IMD_n .	90

3.2.3.1	Définition	90
3.2.3.2	Expression.....	90
3.2.3.3	Bornes.....	90
3.2.3.4	Exemple	91
3.2.4	Résultats, analyse et discussions.....	92
3.3	Métriques d'innovation (MI)	94
3.3.1	Enjeux	94
3.3.2	Métrique d'innovation (MI).....	94
3.3.3	Indicateur d'innovation (II_n)	95
3.3.4	Bornes.....	97
3.3.5	Exemple	97
3.3.6	Résultats et discussion.....	98
3.4	Métrique de Mise en Œuvre d'un Projet Mécatronique (MMOPM).....	99
4	Expérimentation.....	101
4.1	Description du système	101
4.1.1	Un système mécatronique intégré.....	101
4.1.2	Modélisation hiérarchique des différentes architectures candidates du système	102
4.1.2.1	Hypothèses/Cahier des charges	102
4.1.2.2	Le moteur-roue ez-Wheel	103
4.1.2.3	Le moteur-roue produit par GEC Alsthom	104
4.1.2.4	Le moteur-roue Active Wheel de chez Michelin	104
4.2	Résultats.....	105
4.2.1	Calcul de la métrique d'intégration multi-niveau : MMLI.....	105
4.2.2	Calcul de la métrique d'intégration multi-domaine : MMDI.....	106
4.3	Analyse des résultats et discussions	109
5	Discussions	112
6	Automatisation du calcul des métriques	115
7	Conclusions et perspectives	121

Chapitre 3 - Métrique géométrique pour l'intégration physique122

1	Introduction.....	123
1.1	Le contexte	123
1.2	Problématique et plan.....	123
2	Etat de l'art.....	124

2.1	Intégration physique des systèmes mécatroniques	124
2.1.1	Définition	124
2.1.2	Une « r »évolution industrielle rapide	125
2.1.3	De nouveaux moyens et méthodes de conception et de fabrication	126
2.1.4	Conséquences sur la conception mécatronique	129
2.2	Modélisation géométrique et placement des composants	130
2.2.1	Modélisation géométrique	130
2.2.2	Positionnement des objets	131
2.3	Evaluation des architectures spatiales	132
2.3.1	En milieu industriel	132
2.3.2	Approche mathématique	132
2.3.2.1	Propriété d'une distance	132
2.3.2.2	Définition de la distance d'Hausdorff	134
2.3.3	Les méthodes et métriques existantes pour l'évaluation de l'intégration physique	136
2.4	Problématique	137
3	Métrique d'intégration physique développée	137
3.1	Indicateurs de compacité	139
3.1.1	Enjeux et définitions	139
3.1.2	Indicateur de compacité réelle	139
3.1.2.1	Expression	139
3.1.2.2	Bornes	140
3.1.3	Indicateur de forme	140
3.1.3.1	Définition	140
3.1.3.2	Expression	141
3.1.4	Indicateur de compacité convexe	141
3.1.4.1	Expression	141
3.1.4.2	Bornes	142
3.2	Indicateur d'accessibilité	143
3.2.1	Définition	143
3.2.1.1	Expression	143
3.2.1.2	Bornes	145
3.2.1.3	Exemples	145
3.3	Métrique d'intégration physique (MIP)	145
3.3.1	Expression	145
3.3.2	Discussion	146

4	Exemple mécatronique	146
4.1	Présentation du système et des différentes architectures.....	147
4.1.1	Le système hybride série	147
4.1.2	Le système hybride parallèle	147
4.1.3	Le système hybride mixte	148
4.1	Mesure de la compacité réelle.....	149
4.1.1	Le système hybride série	149
4.1.2	Le système hybride parallèle	150
4.1.3	Le système hybride mixte	152
4.1.4	Synthèse.....	153
4.2	Indicateur de forme.....	153
4.3	Indicateur de compacité convexe	154
4.4	Indicateur d'accessibilité	154
5	Implémentation des données géométriques en SysML.....	155
5.1	SysML et les profils	156
5.2	Définition de l'extension SysML pour la géométrie	156
5.3	Applications aux architectures spatiales de la motorisation hybride	159
5.4	Placement des composants.....	161
6	Discussion/perspectives	162
7	Conclusion	162

Chapitre 4 - Processus de création de métriques personnalisées.....164

	Introduction.....	165
1.1	Contexte industriel	165
1.2	Le concept	166
1.3	Règles de formalisation d'une métrique.....	168
2	Description du processus	168
2.1	L'analyse des besoins	168
2.2	Les objectifs à évaluer	169
2.2.1	Les objectifs de conception	169
2.2.2	Les intérêts de ces objectifs pour la mécatronique	170
2.3	Identification des paramètres influents.....	171
2.4	Construction de la métrique	172
2.5	Validation de la métrique.....	173
2.6	Analyse de la sensibilité	173

3	Application de la méthode	175
3.1	L'analyse du besoin	175
3.2	Définition de l'objectif.....	175
3.3	Identification des paramètres influents.....	176
3.4	Construction de la métrique	176
3.4.1	Hehenberger & al. (2005) : modularité fonctionnelle.....	176
3.4.2	Zhang et al. (2003) : modularité pour le cycle de vie (et coût)	177
3.4.3	Newcomb & al. (1998) : modularité de conception pour le cycle de vie.....	177
3.4.4	Turki (2008) : Métrique de modularité des blocks.....	178
3.5	Validation de la métrique.....	179
4	Discussion et perspectives	179
5	Conclusion	181
Conclusion générale.....		182
Constat		183
Evaluation d'architectures		183
Conception des systèmes mécatroniques.....		183
Besoin		183
Contributions.....		184
Perspectives.....		185
Bibliographie		187
Annexes.....		204
Annexe 1 : Recherche de concept.....		205
Annexe 2 : Recherche et émergence d'architecture candidates		207
Annexe 3 : Formalisation / modélisation des architectures		209
Annexe 4 : Modélisation hiérarchique		215
Annexe 5 : Définition mathématique.....		218
Annexe 6 : Construction d'une enveloppe convexe 3D		220
Annexe 7 : Programme de construction d'une enveloppe convexe 3D en langage JAVA.....		222

Liste des figures

Introduction

Figure 0-1 : Définition de la mécatronique	2
Figure 0-2 : Intégration fonctionnelle et physique d'un produit mécatronique (NF 2008).....	3

Chapitre 1

Figure 1-1 : Exemples de systèmes mécatroniques	7
Figure 1-2 : Comparaison des cycles de vie (INCOSE 2006; Hammadi 2012a)	9
Figure 1-3 : Représentation des différentes étapes de la phase de conception	10
Figure 1-4 : Le développement en cascade (Tahan et al. 2011)	10
Figure 1-5 : Le cycle en V (Tahan et al. 2011).....	11
Figure 1-6 : Cycle de conception en spirale (Boehm 1988).....	11
Figure 1-7 : La méthode 2TUP (Tahan et al. 2011).....	12
Figure 1-8 : Méthode de développement en X.....	13
Figure 1-9 : Les différents diagrammes SysML.....	17
Figure 1-10 : Niveaux de modélisation de la conception et architectures associées.....	19
Figure 1-11 : Exemple de l'évolution mécatronique d'un disjoncteur - Integral 32 et Tesys U (Schneider Electric).....	22
Figure 1-12 : Intégration multi-domaine : l'alternateur i-Stars (Valeo®).....	22
Figure 1-13 : Diagramme d'exigences (req) (Mhenni et al. 2014)	24
Figure 1-14 : Diagramme du cycle de vie (stm) (Mhenni et al. 2014).....	25
Figure 1-15 : Diagramme de contexte (bdd) (Mhenni et al. 2014)	25
Figure 1-16 : Diagramme d'interfaces externes (ibd) (Mhenni et al. 2014)	26
Figure 1-17 : Diagramme de modes de fonctionnements du système (stm) (Mhenni et al. 2014).....	26
Figure 1-18 : Diagramme de cas d'utilisation (ucd) (Mhenni et al. 2014).....	26
Figure 1-19 : Diagramme de séquence (seq) (Mhenni et al. 2014)	27
Figure 1-20 : Diagramme de traçabilité des exigences	27
Figure 1-21 : Diagramme d'activités (act) (Mhenni et al. 2014)	28
Figure 1-22 : Diagramme de définition de bloc décrivant une architecture logique (Mhenni et al. 2014).....	28
Figure 1-23 : Exemple de diagramme interne de définition de blocs pour l'architecture logique (Mhenni et al. 2014)	29
Figure 1-24 : Exemple de diagramme paramétrique (Mhenni et al. 2014)	29
Figure 1-25 : Schématisation de la définition d'une métrique à partir de données de la littérature	32
Figure 1-26 : Les métriques dans le cycle de conception.....	35
Figure 1-27 : Niveaux de probabilité de risques d'un évènement (Ferguson 2004).....	37
Figure 1-28 : Niveaux de valeur d'impact d'un risque (Ferguson 2004)	37
Figure 1-29 : Identification non exhaustive des paramètres influents dans l'évaluation de la fiabilité (Zwingmann 2005).	43
Figure 1-30 : Exemple d'échelle pour quelques critères (Das et al. 2000a).....	46

Figure 1-31 : Restriction d'accessibilité sur certaines directions en présence d'autres composants (Hsu and Lin 2002)	47
Figure 1-32 : Volume accessible d'un composant et détail de la décomposition de ce volume (Hsu and Lin 2002).....	47
Figure 1-33 : Degré de liberté et direction du composant (Fujimoto et al. 2001).....	49
Figure 1-34 : Espace nécessaire pour le retrait des pièces (Fujimoto et al. 2001)	49
Figure 1-35 : Synthèse des différentes méthodes pour évaluer l'effort de désassemblage (Zwingmann 2005).....	51
Figure 1-36 : Paramètres de recyclabilité (Atlee and Kirchain 2006)	52
Figure 1-37 : Exemple d'indicateurs TPM (Oakes et al. 2006)	54

Chapitre 2

Figure 2-1 : Processus de conception architecturale vu par l'INCOSE (INCOSE 2006)	61
Figure 2-2 : Diversité des activités de l'architecte système (Muller 2011).....	62
Figure 2-3 : Démarche générale de la méthode TRIZ (Dubois 2004).....	64
Figure 2-4 : Architecture fonctionnelle (actigramme) et organique (datagramme) avec SADT/IDEF ⁴⁶	65
Figure 2-5: Principe des architectures FAST	65
Figure 2-6 : Structure hiérarchique d'un système mécatronique (Kellner and Hehenberger 2014).....	66
Figure 2-7 : Processus d'aide à la décision (Ullman 2000)	67
Figure 2-8 : Maison de la qualité HoQ I	68
Figure 2-9 : Les quatre maisons de la qualité (Towar 2008)	70
Figure 2-10: Exemple de DSM.	70
Figure 2-11 : Exemple d'une représentation DMM entre les exigences clients et le cahier des charges du produit (Danilovic and Browning 2007).....	71
Figure 2-12 : DSM et DMM spécifiques pour le système et le produit (Danilovic and Browning 2007).....	71
Figure 2-13 : Les quatre domaines de l'Axiomatic Design (Park 2007).....	72
Figure 2-14 : Développement des métriques d'intégration multi-domaine et multi-niveau..	78
Figure 2-15 : Premier exemple d'architecture candidate	82
Figure 2-16 : Deuxième exemple d'architecture candidate.....	82
Figure 2-17 : Schématisation des termes de la formule (2.5).....	83
Figure 2-18 : Résultats des deux architectures candidates en fonction des indicateurs.....	85
Figure 2-19 : Les différents niveaux d'intégration dans les systèmes mécatronique (Projet O2M : Outils de Modélisation Mécatronique)	87
Figure 2-20 : Exemples théoriques avec enrichissement des domaines des éléments.....	91
Figure 2-21 : Synthèse des résultats pour l'intégration multi-domaine des deux architectures candidates	93
Figure 2-22 : Exemple théorique avec enrichissement des éléments en fonction de leur valeur d'innovation	97
Figure 2-23 : Modélisation hiérarchique du système ez-Wheel	103
Figure 2-24 : Moteur-roue créé par Alstom pour l'Irisbus Cristalis	104
Figure 2-25: Modélisation hiérarchique du moteur-roue "Active Wheel" de chez Michelin	105
Figure 2-26 : Décomposition par domaine pour l'architecture utilisant le système ez-Wheel	106

Figure 2-27 : Décomposition par domaine pour l'architecture utilisant le système Gec	107
Figure 2-28 : Décomposition par domaine pour l'architecture utilisant le système Active Wheel	108
Figure 2-29 : Récapitulatif des indicateurs d'intégration multi-niveau pour chaque architecture et pour chaque niveau d'abstraction	109
Figure 2-30: Résultat des indicateurs d'intégration multi-domaine pour chaque architecture et pour chaque niveau d'abstraction	110
Figure 2-31 : Analyse des solutions candidates à l'aide des métriques d'intégration multi-domaine et multi-niveau	111
Figure 2-32 : Exemple de diagramme en étoiles associé à six indicateurs.	112
Figure 2-32 : Exemple d'éléments des architectures pour chaque niveau.....	116
Figure 2-33 : Identifiant des éléments de chaque diagramme et identifiants des différents digrammes.....	117
Figure 2-34 : En-tête du fichier.....	118
Figure 2-35 : Calcul des indicateurs pour l'architecture 1	118
Figure 2-36: Calcul des indicateurs pour l'architecture 2	119
Figure 2-37 : Calcul des indicateurs pour l'architecture 3	119
Figure 2-38 : Calcul des indicateurs d'intégration multi-domaine pour l'architecture 1.....	120
Figure 2-39 : Calcul des indicateurs d'intégration multi-domaine pour l'architecture 2.....	120
Figure 2-40 : Calcul des indicateurs d'intégration multi-domaine pour l'architecture 3.....	120

Chapitre 3

Figure 3-1 : Intégration physique d'un produit mécatronique	125
Figure 3-2 : Exemples de systèmes mécatroniques très intégrés physiquement.....	125
Figure 3-3 : Active Wheel de Michelin	126
Figure 3-4 : Une haute intégration du système (source : Projet O2M ³).....	126
Figure 3-5 : Exemple d'applications MID remplaçant le câblage conventionnel d'un volant de direction multifonction d'une voiture (Heininger et al. 2014).....	127
Figure 3-6 : Module de puissance 3D de chez Valeo ³⁷	127
Figure 3-7 : Module de puissance du convertisseur d'un GMPE (Hammadi 2012)	128
Figure 3-8 : Illustration des avantages de l'intégration physique sur un système mécatronique : le roulement à billes instrumenté (SKF™)	129
Figure 3-9: Distances de Manhattan (chemins rouge, jaune et bleu) et Distance euclidienne (en vert)	133
Figure 3-10 : Exemple de deux ensembles : un carré et un disque	134
Figure 3-11 : Représentation des distances d'Hausdorff a et b	135
Figure 3-12 : Distance d'Hausdorff entre la ligne verte X et la ligne bleue Y.....	135
Figure 3-13 : Définition de la demi-distance d'Hausdorff (Jaulin et al. 2001)	136
Figure 3-14 : Les métriques dans le cycle de conception.....	138
Figure 3-15 : Les différents indicateurs de la métrique d'intégration physique.....	138
Figure 3-16 : Assemblage de 3 composants S1, S2 et S3	140
Figure 3-17 : Assemblage des composants S1, S2 et S3 dans un volume disponible cubique	141
Figure 3-18 : Illustration d'ensemble complexe et non convexe	142
Figure 3-19 : Enveloppe convexe X d'un sous-ensemble X	142
Figure 3-20 : Illustration d'un volume de passage	143

Figure 3-21 : Illustration en 2D du volume du passage disponible	144
Figure 3-22 : Explication de r_{objet}	145
Figure 3-23 : Système hybride série (Toyota Motor Corporation 2003)	147
Figure 3-24 : Système hybride parallèle (Toyota Motor Corporation 2003)	148
Figure 3-25 : Système hybride mixte (Toyota Motor Corporation 2003)	148
Figure 3-26 : Architecture d'un système hybride série	150
Figure 3-27 : Extraction des données de surface et de volume sous Catia pour le Système Hybride Série	150
Figure 3-28 : Architecture d'un système hybride parallèle.....	151
Figure 3-29 : Extraction des données de surface et de volume pour le système hybride parallèle.....	151
Figure 3-30 : Architecture d'un système hybride mixte.....	152
Figure 3-31 : Extraction des données de surface et de volume du système hybride mixte..	152
Figure 3-32 : Composants enrichis de paramètres géométriques.....	157
Figure 3-33 : Librairie des composants géométriques.....	158
Figure 3-34 : Exemple de composants géométriques avec leurs paramètres.....	158
Figure 3-35 : Création d'un bloc géométrique	159
Figure 3-36 : Choix d'une forme spécifique sur un composant existant stéréotypé « geometry »	159
Figure 3-37 : Architecture du système hybride série.....	160
Figure 3-38 : Architecture du système hybride parallèle.....	160
Figure 3-39 : Architecture du système hybride mixte.....	161
Figure 3-40 : Diagramme paramétrique permettant de calculer la métrique lcr.....	161
Figure 3-41 : Utilisation de la distance d'Hausdorff pour le placement des composants sous l'influence physique d'un autre composant	162

Chapitre 4

Figure 4-1: Processus de "l'usine à métriques"	167
--	-----

Liste des tables

Chapitre 1

Table 1-1: Tableau récapitulatif des termes de la définition d'une métrique	33
Table 1-2 : Liste de critères pour évaluer la maintenabilité (Wani and Gandhi 1999).....	45
Table 2-1 : Résultats des indicateurs d'intégration multi-niveau	105

Chapitre 2

Table 2-2 : Résultat de la métrique d'intégration multi-niveau.....	106
Table 2-3 : Résultat des indicateurs d'intégration multi-domaine IMD pour chaque niveau d'abstraction.....	108
Table 2-4 : Résultats du calcul de la métrique d'intégration multi-domaine mécatronique.	109
Table 2-5 : Tableau récapitulatif des résultats des métrique d'intégration multi-domaine et multi-niveau	111

Chapitre 3

Table 3-1 : Les 7 classes de surface et leurs EGRM (Clément et al. 1998).....	131
Table 3-2 : Les 13 classes de contraintes entre SATT (Clément et al. 1998; Choley 2005)....	132
Table 3-3: Dimensions des composants communes aux 3 architectures	149
Table 3-4 : Dimensions et formes des composants du système hybride série.....	149
Table 3-5 : Dimensions et formes des composants du système hybride parallèle.....	151
Table 3-6 : Dimensions et formes des composants du système hybride mixte	152
Table 3-7 : Synthèse des résultats de calculs de l'indicateur de compacité réelle.....	153

Chapitre 4

Table 4-1 : Quelques objectifs pour la mécatronique.....	170
--	-----

INTRODUCTION

Le contexte général des travaux de recherche présenté dans ce manuscrit est la conception des systèmes mécatroniques dans les phases de préconception.

L'objectif global porte plus particulièrement sur l'aide à l'évaluation d'architectures suivant une approche d'Ingénierie Systèmes appliquée ici à la conception des systèmes mécatroniques.

1 Problématique

Le terme « mécatronique » a été introduit pour la première fois en 1969 par un ingénieur de la société japonaise Yaskawa Electric Corporation. La définition de la mécatronique est donnée dans la norme NF E01-010 (NF 2008) comme étant une « démarche visant l'intégration en synergie de la mécanique, l'électronique, l'automatique et l'informatique dans la conception et la fabrication d'un produit en vue d'augmenter et/ou d'optimiser sa fonctionnalité ».

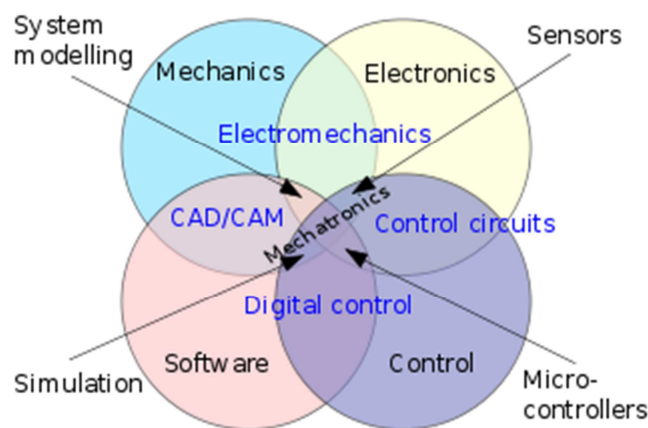


Figure 0-1 : Définition de la mécatronique

Aujourd'hui, pour être compétitifs, les industriels doivent être en mesure de valider leur système dans les premières étapes de conception. Selon une enquête de Aberdeen Group (Boucher and Houlihan 2008) sur les besoins industriels actuels des produits mécatroniques, les clients souhaitent des produits moins chers, variés avec une meilleure qualité et fiabilité, tout en étant toujours plus "intelligents" ou innovants à forte valeur ajoutée, tandis que le temps de développement doit être de plus en plus court. Ainsi, le processus de développement des produits mécatroniques est contraint d'évoluer pour répondre à ces nouvelles exigences du marché.

La réalisation d'un système mécatronique de nature hétérogène met en œuvre plusieurs métiers concevant et réalisant des constituants de natures différentes qu'il faut assembler. Actuellement, trop de processus de conception mécatronique sont uniques, dépendant de l'expérience ou de l'intuition d'ingénieurs seniors, sans aucun support méthodologique ou de validation. Comme l'un des enjeux importants de la conception des systèmes mécatroniques est de gérer les nombreuses données de conception communes aux concepteurs et architectes des diverses disciplines d'ingénierie impliquées, il est essentiel

que la modélisation de ces systèmes fournisse des informations prédictives au plus tôt dans le cycle de vie. En outre, un des enjeux clé de ces systèmes se focalisent la plupart du temps autour du concept d'intégration. En effet, les propriétés d'un système ne résultent pas des seules propriétés de ses composants, mais sont également fonction de leurs interactions. Les problèmes d'intégration de ces composants et de maîtrise de leur hétérogénéité sont donc devenus primordiaux. Ainsi, le concepteur mécatronicien doit habituellement traiter les trois objectifs d'intégration suivant : fonctionnelle, multi-domaine et physique. La norme NF E01-010 (NF 2008) définit explicitement deux axes d'intégration : fonctionnelle et physique. Selon la norme, l'intégration fonctionnelle exprime l'intégration « des fonctions de détection, de communication et de traitement de l'information aux fonctions mécaniques de base ». L'intégration physique concerne « l'interpénétration des supports mécaniques et électroniques qui embarquent les fonctions automatiques et informatiques ».

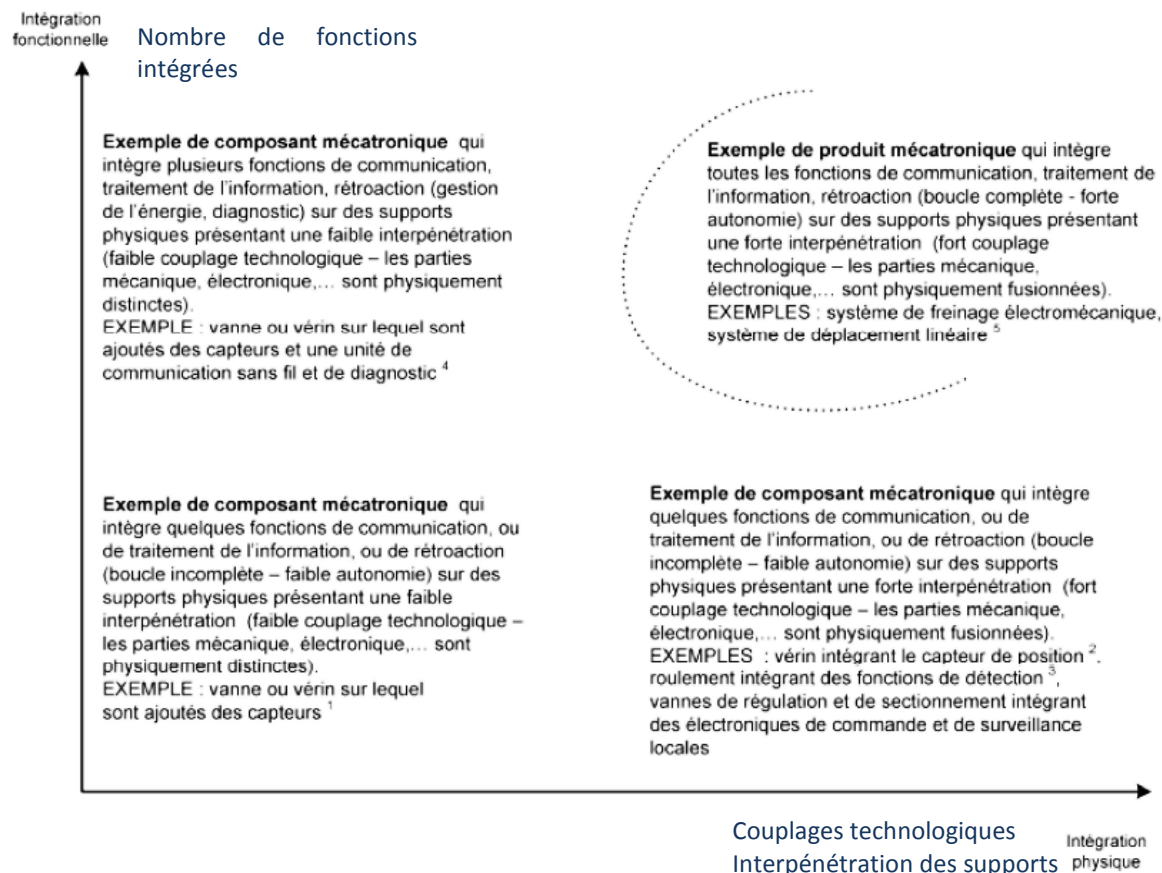


Figure 0-2 : Intégration fonctionnelle et physique d'un produit mécatronique (NF 2008)

De plus, la conception de systèmes mécatroniques génère, à partir des spécifications multidisciplinaires, un espace de solutions possibles si large, que les concepteurs ont besoin de moyens efficaces pour sélectionner la solution optimale au sein de cet espace, qui devra satisfaire les nombreuses contraintes issues du domaine de la mécatronique.

Jusqu'à présent l'évaluation de ces intégrations est une évaluation qualitative qui permet de comparer plusieurs produits mécatroniques en les triant en fonction de leur niveau d'intégration. La problématique est de pouvoir trouver la ou les « bonnes » architectures qui satisfassent les objectifs du client et aident les concepteurs dans leurs choix tout en assurant la traçabilité de ces derniers et ainsi de pouvoir comparer quantitativement les

différentes architectures. L'idée est donc de proposer des critères de choix et un moyen de les mesurer pour contrôler l'adéquation des architectures avec les objectifs du concepteur. Pour cela, nous proposons d'utiliser des métriques. En effet, les métriques sont destinées en général à évaluer la conception des systèmes et la qualité des modèles pour aider le concepteur à prendre des décisions, détecter et vérifier les défauts, au plus tôt dans le cycle de conception.

2 Structure du mémoire

Le **Chapitre 1** intitulé « Contexte et problématique de l'évaluation en conception mécatronique », est consacré à la définition de la mécatronique et de ses enjeux, en s'appuyant sur la notion d'**intégration** qui est un objectif clé dans la conception des systèmes mécatroniques. Pour aider les concepteurs et l'architecte système à faire le meilleur choix parmi plusieurs solutions candidates et ainsi leur permettre de justifier et tracer leurs choix, l'utilisation des métriques a été choisie. En effet, les métriques sont un support d'aide à la décision intéressant pour aider à formaliser et garantir une traçabilité des critères sur lesquels les décisions finales seront prises. Dans ce contexte, les règles de définition des métriques sont présentées ainsi qu'un état de l'art sur des métriques existantes, tout au long du cycle de conception. Cela nous amène à définir notre problématique en lien avec le traitement des métriques dans les phases préliminaires de conception pour répondre à l'évaluation des systèmes mécatroniques selon l'objectif d'intégration.

Le **Chapitre 2** intitulé « Métriques d'architectures pour l'intégration multi-domaine et multi-niveau », présente dans un premier temps, un état de l'art sur les architectures, de leurs définitions aux aides à la décision pour leur choix, en passant par leur cycle de vie au cours de la conception.

Par la suite, les métriques d'intégration offrant la possibilité de mesurer quantitativement le niveau d'intégration multi-niveau et multi-domaine des différentes architectures candidates seront développées. Au regard, de la nécessité pour les concepteurs de pouvoir disposer en phase de préconception, d'une solution d'architecture répondant au mieux à l'objectif d'intégration multi-domaine et multi-niveau, parmi plusieurs architectures candidates.

D'autres métriques d'innovation et de mise en œuvre d'un projet mécatronique seront présentées en complément des métriques précédentes. Nous montrons que l'ensemble de ces métriques permettent aux concepteurs et à l'architecte système de choisir l'architecture correspondant aux mieux à ces objectifs, de justifier et tracer leurs choix.

Ce chapitre s'achève avec une illustration de ces métriques sur un système mécatronique intégré : « le moteur-roue », suivi par une présentation de l'automatisation de ces calculs à l'aide de l'outil Artisan Studio de la société Atego pour la modélisation SysML des architectures et à une extension du profile SysML.

Le **Chapitre 3**, intitulé « Métriques géométriques pour l'intégration physique » se focalise sur les métriques d'intégration géométrique et spatiale. En effet, la mécatronique a permis d'augmenter la compacité des objets de notre quotidien, tout en augmentant leur intégration géométrique. Cependant avec des volumes d'encombrements de plus en plus petits, il devient primordial de prendre en compte le placement des composants pour

optimiser l'assemblage des sous-systèmes. De plus, la proximité induite des composants peut, à la fois amener la création de nouvelles fonctions permettant d'augmenter les performances du système global, mais aussi parfois causer des problèmes lorsque les couplages multi-physiques causent des dommages aux composants environnants. Aussi, pour évaluer cette intégration physique, une métrique a été développée en utilisant deux indicateurs : l'indicateur de densité et l'indicateur d'accessibilité. Par ailleurs, pour pouvoir calculer ces métriques géométriques au plus tôt dans le cycle de conception (en préconception), nous avons développé une extension SysML pour la géométrie. Enfin, ces métriques ont été illustrées sur un exemple d'application d'un système mécatronique : la motorisation hybride d'un véhicule automobile. Les outils numériques utilisés sont le logiciel Artisan Studio pour la modélisation SysML et l'extension géométrique du profil SysML ainsi que le logiciel de CAO Catia de la société Dassault Systèmes pour le calcul des enveloppes des systèmes mécatroniques.

Finalement, compte tenu qu'actuellement, l'organisation de chaque entreprise dépend fortement des méthodes de conception de leurs concepteurs sans support méthodologique ou de validation, le **Chapitre 4** intitulé « Processus de création de métriques personnalisées » propose une démarche permettant de construire des métriques personnalisées, en vue d'évaluer instantanément et ce pour un objectif donné, un processus, un projet, une conception d'un système et une qualité du produit. Ces spécificités sont particulièrement fondamentales au regard de la complexité de la conception des systèmes mécatroniques. Ce processus sera illustré par un exemple répondant à un objectif souvent lié à celui de l'intégration : la modularité.

Le mémoire se termine par quelques conclusions et perspectives.

Chapitre 1

Contexte et problématique de l'évaluation en conception mécatronique

1	Introduction.....	7
1.1	La mécatronique.....	7
1.2	La conception des systèmes.....	8
2	L'intégration dans la conception mécatronique	20
2.1	Intégration et complexité des systèmes mécatroniques.....	21
2.2	Spécificités de la conception mécatronique et méthode associée.....	23
2.3	Enjeux pour l'évaluation de la conception mécatronique	30
3	Aide aux choix de conception mécatronique : les métriques.....	30
3.1	Intérêt des métriques.....	30
3.2	Définition des métriques.....	32
4	Métriques dans le cycle de conception.....	34
4.1	Les métriques en phase de préconception	35
4.2	Les métriques dans la phase de conception préliminaire.....	40
4.3	Les métriques dans la phase de conception détaillée	45
4.4	Les métriques en phase de vérification et validation	53
5	Conclusions.....	56

1 Introduction

L'apparition des systèmes mécatroniques depuis une vingtaine d'années peut être considérée comme une révolution pour le monde industriel. En effet, l'utilisation de ces systèmes s'est rapidement généralisée et influence actuellement la quasi-totalité des secteurs de l'industrie.

En parallèle, la conception de ces systèmes est devenue de plus en plus complexe du fait de leur caractère pluri-technologique (Bishop 2007; Penas et al. 2009). Ainsi après avoir défini le terme « mécatronique » et avoir rappelé les principaux cycles de conception, nous nous focaliserons sur l'un des problèmes majeurs rencontrés lors de la phase de conception des systèmes mécatroniques : l'évaluation de cette conception.

Pour répondre à cette problématique, nous nous sommes penchés sur une méthode d'évaluation qui est l'utilisation de métriques.

Ce premier chapitre se terminera donc par un état de l'art sur les métriques tout au long du cycle de conception des systèmes.

1.1 La mécatronique

La première utilisation du mot mécatronique (mechatronic en anglais) a été mise en œuvre par un ingénieur de la compagnie japonaise Yaskawa Electric Corporation en 1969 pour désigner le contrôle des moteurs électriques par ordinateur (Mihalache 2007; Hammadi 2012a). Initialement, le terme mécatronique était un néologisme rassemblant les termes "méca" pour "mécanique" et "tronique" pour "électronique" (Turki 2008a).

Depuis, plusieurs définitions sont apparues, correspondant mieux à l'évolution des systèmes mécatroniques actuels (Figure 1-1), comme les véhicules automobiles modernes, l'ABS, l'ESP, ou des objets de la vie quotidienne comme les robots aspirateurs, ou encore les roulements à billes instrumentés,...



Figure 1-1 : Exemples de systèmes mécatroniques

Nous citons ci-dessous, de manière non exhaustive, certaines d'entre-elles. A titre d'exemple, le journal international *Mechatronics*¹ définit la mécatronique *comme étant la combinaison synergétique de l'ingénierie mécanique de précision, du contrôle électronique et l'approche systémique dans la conception des produits et les processus de fabrication. Elle concerne la conception des systèmes, appareils et produits visant à atteindre un compromis optimal entre la structure mécanique basique et son contrôle.* Une autre définition est

¹ <http://www.journals.elsevier.com/mechatronics/>

donnée par le journal IEEE/ASME Transactions on Mechatronics² : *la mécatronique est l'intégration synergétique de l'ingénierie mécanique avec une électronique et un contrôle informatique intelligent dans la conception et la fabrication des produits et processus industriels.*

Ce n'est que bien plus tard, en novembre 2008, que la norme française NF E01-010 (NF 2008) a donné sa première définition de la mécatronique : « *démarche visant l'intégration en synergie de la mécanique, l'électronique, l'automatique et l'informatique dans la conception et la fabrication d'un produit en vue d'augmenter et/ou d'optimiser sa fonctionnalité* », en précisant que « *l'objectif de la mécatronique est l'obtention d'une valeur ajoutée supérieure à la simple somme des valeurs ajoutées des fonctions prises séparément* ».

En effet, en conception mécatronique, il ne s'agit pas seulement d'assembler des composants de domaines technologiques différents, mais il est nécessaire de considérer le système dans sa globalité pendant tout le cycle de conception (Turki 2008). Dans la section suivante, nous nous intéressons à ces différents cycles de conception.

1.2 La conception des systèmes

Espanet (Espanet 1999a) définit la conception de la manière suivante : *"la conception correspond à la définition d'un objet ou d'un système (ensemble d'objets) répondant à un besoin défini dans un cahier des charges"*.

Etant entendu que le cahier des charges doit contenir la ou les fonctions à réaliser et les contraintes qui reposent sur ces fonctions. Pour cela, la conception doit prendre en compte toutes les contraintes émergeant du cycle de vie du système, en adéquation avec la démarche d'ingénierie système. En effet, le cycle de vie correspond à une vision séquentielle des phases de la vie du système. Il s'appuie sur une « organisation phasée des activités qui jalonnent la vie du système depuis l'émergence de son besoin jusqu'à son retrait de service ». L'INCOSE (International Council on Systems Engineering) (INCOSE 2006) a ainsi réalisé une étude comparative des cycles de développement des systèmes suivant les différentes normes (Figure 1-2).

² <http://ieeexplore.ieee.org/xpl/aboutJournal.jsp?punumber=3516#AimsScope>

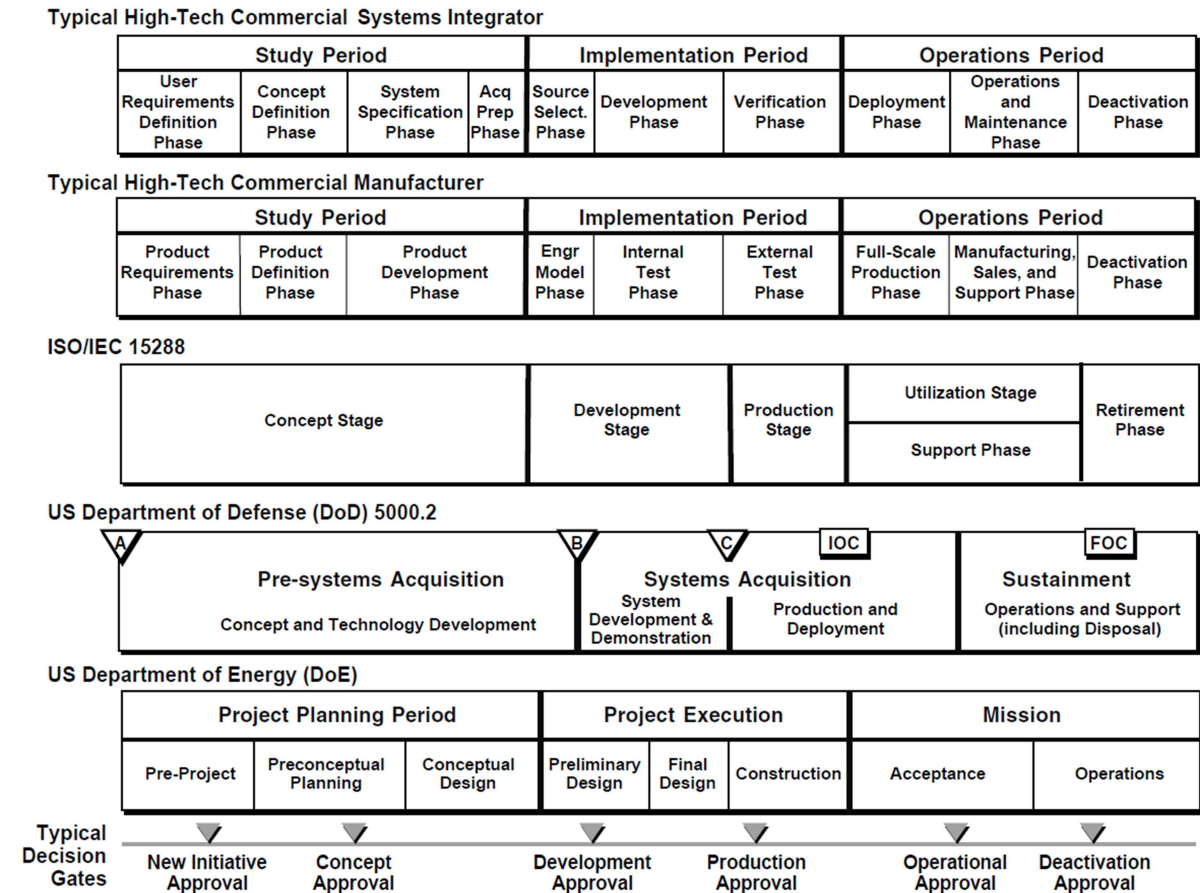


Figure 1-2 : Comparaison des cycles de vie (INCOSE 2006; Hammadi 2012a)

Globalement, les phases principales identifiées sont la conceptualisation, le développement, la production/réalisation, (l'installation), l'utilisation et le retrait de service (Meinadier 1998). Selon la norme ISO/IEC-15288, le processus de conception des produits est constitué de deux étapes : l'étape des concepts (Concept stage) et l'étape de développement.

L'étape des concepts, selon le US Department of Energy (DoE), est constituée des phases : avant-projet, planification pré-conceptuelle et le 'conceptual design'. L'étape de développement est constituée d'une phase de conception préliminaire et d'une phase de conception détaillée.

Au final, le processus de conception dans ces phases est basé sur l'élaboration des modèles, leur vérification et leur validation (Hammadi 2012a).

1.2.1 Les principaux cycles de conception

Sur la base de la norme ISO/IEEE 15288 :2001 (ISO 2001), nos travaux se situent dans la conception des systèmes intégrant la phase de préconception (« Concept stage ») et de développement. Nous avons représenté ces différentes étapes dans l'illustration ci-dessous (Figure 1-3) :

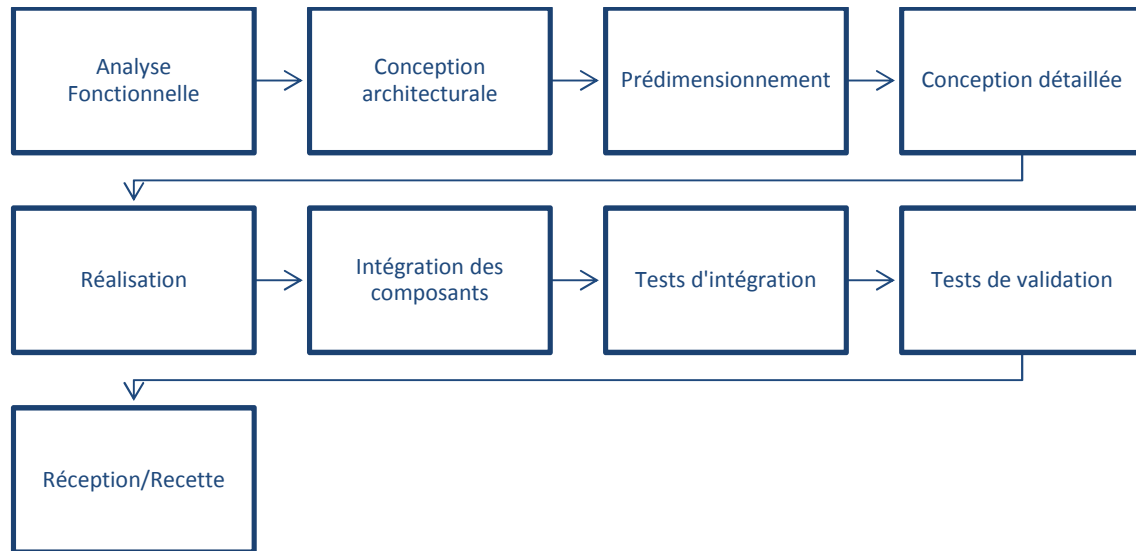


Figure 1-3 : Représentation des différentes étapes de la phase de conception

En s'appuyant sur la synthèse de Tahan et al. (Tahan et al. 2011), voici une présentation des principaux cycles de conception, qui se termine par le cycle en X (cycle qu'ils ont développé).

1.2.1.1 Le cycle de conception en cascade

Le premier cycle de conception est celui du *développement en cascade* (Figure 1-4). Il a été défini par Winston W. Royce en 1970, et est couramment utilisé dans l'industrie du BTP. A partir d'une idée, les activités de conception se déroulent les unes après les autres jusqu'à la réalisation. L'axe vertical représente le degré d'avancement et l'axe horizontal représente l'évolution de la conception dans le temps.

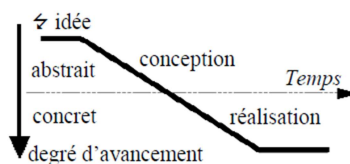


Figure 1-4 : Le développement en cascade (Tahan et al. 2011)

L'avantage est sa simplicité mais il ne convient pas à la conception de systèmes plus complexes car la conception de ces systèmes n'est pas séquentielle, il est souvent difficile de définir tous les besoins dès le début du projet, et la validation de la conception avec cette approche est tardive (Med Amine Ghodhbane 2012).

1.2.1.2 Le cycle de conception en V

Le *cycle en V* est aujourd'hui le cycle le plus connu et le plus utilisé dans la conception de systèmes. Il a été développé en 1986 par Goldberg, et trouve son origine dans la conception de logiciels afin de pallier les problèmes de réactivité du modèle en cascade (Hammadi 2012a). En effet, le cycle en V permet de détecter très tôt d'éventuelles anomalies et donc de limiter le retour aux phases précédentes. Il se compose de deux parties : une partie descendante analogue au développement en cascade et une partie montante.

La partie montante permet une analyse et une vérification avant l'intégration des composants : elle renvoie les spécifications de la partie descendante sur les phases en vis-à-vis de la partie montante.

De la même manière, les phases de la partie montante signalent les défauts détectés/écarts aux spécifications pour chaque phase de la partie descendante (Figure 1-5). L'axe vertical représente le niveau de détail et l'axe horizontal représente l'avancement dans le temps.

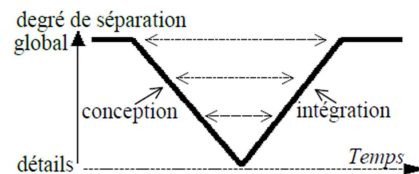


Figure 1-5 : Le cycle en V (Tahan et al. 2011)

1.2.1.3 Le cycle de conception en spirale

Le cycle en spirale est un cycle de conception développé par B. Boehm en 1988 (Boehm 1988). Ce cycle est divisé en quatre quadrants (Figure 1-6) dénommés :

1. Détermination des objectifs, des alternatives et des contraintes;
2. Analyse des risques, évaluation des alternatives ;
3. Développement et vérification de la solution retenue ;
4. Revue des résultats et vérification du cycle suivant.

Il reprend les différentes phases du cycle en V mais en implémentant des versions successives. L'avantage de ce cycle est d'avoir progressivement un produit (système) de plus en plus complet, au fur et à mesure que les informations sont disponibles. Cependant, les itérations augmentent aussi la complexité de conception.

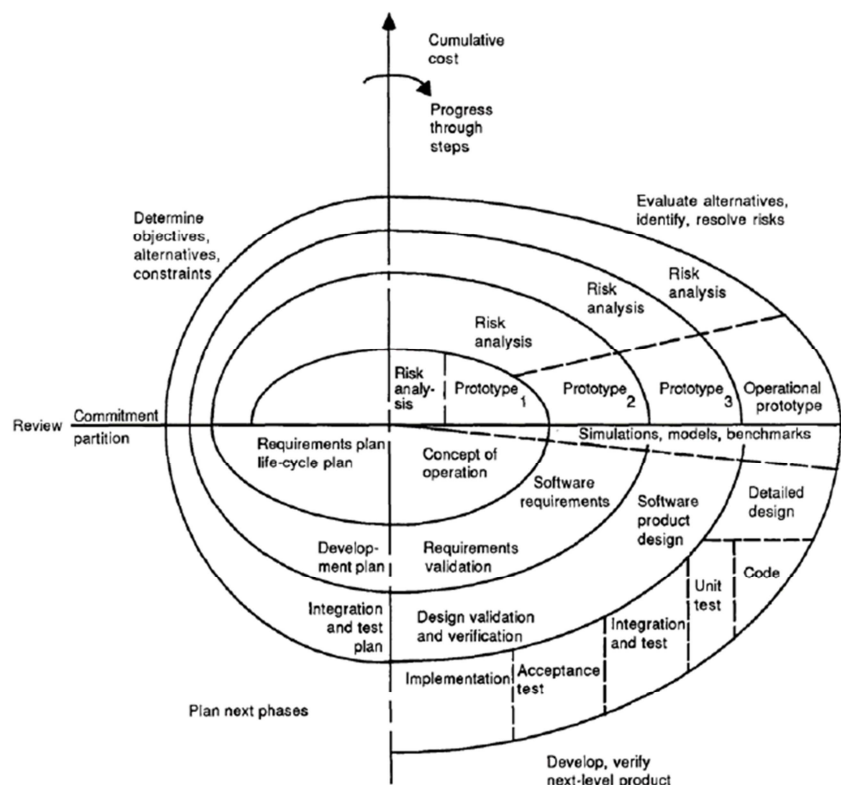


Figure 1-6 : Cycle de conception en spirale (Boehm 1988)

1.2.1.4 Le cycle de conception en Y

Le cycle en Y ou encore appelé 2TUP (Two Tracks Unified Process) proposé plus récemment par la société VALTECH³ au début des années 2000, dissocie les aspects techniques des aspects fonctionnels. Le processus s'articule autour de trois phases (Figure 1-7):

- Une branche fonctionnelle,
- Une branche technique,
- Une branche de réalisation.

Les deux premières sont abordées simultanément durant la phase de capture des besoins et la phase d'analyse. La dernière branche consiste à réunir les deux branches précédentes, permettant de mener la réalisation et la livraison du système (ou produit). Dans ce cycle, l'évolution du temps est représentée selon l'axe vertical.

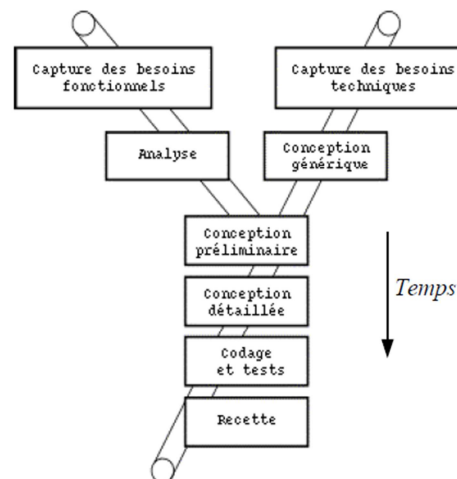


Figure 1-7 : La méthode 2TUP (Tahan et al. 2011)

1.2.1.5 Le cycle de conception en X

Pour finir, la méthode de développement en X, élaborée par le laboratoire Lab-STICC⁴ en 2010, est une synthèse des cycles de conception décrits précédemment, à la différence qu'elle prend en considération l'environnement comme une contrainte de conception présente avant, pendant et après la conception du système.

Le cycle est séparé en deux zones (Figure 1-8) : la première correspond à tous les éléments matériels qui sont au contact de l'environnement (partie basse). La seconde correspond à tout ce qui est immatériel (les idées, les données, les savoirs, les modèles,...) et qui n'ont pas d'effet direct avec l'environnement (partie haute).

Ces deux espaces sont séparés par un axe horizontal représentant l'axe du temps.

³ www.valtech.fr/fr

⁴ <http://www.labsticc.fr/le-pole-cid/>

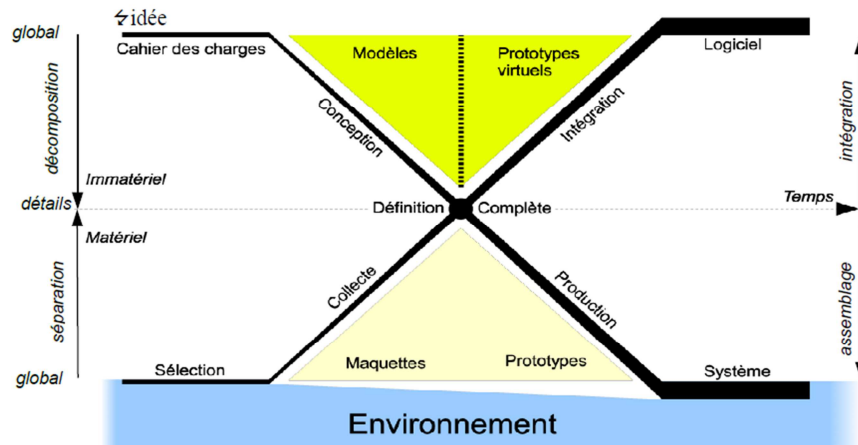


Figure 1-8 : Méthode de développement en X

L'analyse initiale enclenche les deux processus de conception en parallèle. Les flèches d'échanges d'informations nécessaires aux phases en vis-à-vis (comme pour le cycle de conception en V) ne sont plus seulement horizontales mais également verticales (entre le produit virtuel et le produit réel).

Dans les cycles de conception classiques que nous venons de détailler, seul l'axe du temps peut-être quantitatif. Il n'y a, à l'heure actuelle, aucun moyen de mesurer quantitativement l'intégration sur les axes verticaux et horizontaux.

1.2.2 Les méthodes de conception

Maintenant que nous avons présenté les principaux cycles de conception, nous nous intéressons aux méthodes sur lesquelles peuvent s'appuyer ces différents cycles.

1.2.2.1 Démarche séquentielle

Cette démarche, utilisée dans le cycle en V ou de type cascade présente deux principaux types de processus de conception. Il s'agit de la conception dite « Bottom-up » et de la conception dite « Top-down » (Meinadier 1998; Crespi et al. 2008; Hammadi 2012a).

- L'approche « Bottom-up » dispose comme point de départ des spécifications fonctionnelles du système à concevoir, puis les composants de base (résistances, transistors, masse, ressort, etc.) sont ajoutés et liés successivement pour constituer des sous-systèmes et des ensembles plus compliqués. L'avantage de cette méthode est sa simplicité pour construire des structures complexes. Cependant, le système global ne pourra être validé qu'à la fin de la modélisation quand toutes ses parties seront reliées. Ceci a pour conséquence de ne détecter les faiblesses du système que trop tard, après avoir investi un temps considérable en modélisation. La conception classique en mécanique ou en électronique l'utilise couramment.
- L'approche « Top-down » se base sur une démarche de conception allant du plus abstrait au plus détaillé. Le point de départ est un modèle descriptif des attentes de l'utilisateur, puis le modèle est enrichi d'un ensemble de fonctions qui couvre les spécifications du système. Le modèle est ainsi successivement partitionné et raffiné, jusqu'à aboutir à sa définition détaillée. En comparaison de la méthode « Bottom-up », les erreurs et les faiblesses du système sont découvertes au plus tôt car la validation se fait à chaque niveau d'abstraction avant de passer au niveau suivant plus détaillé, ce qui permet de gagner un temps important lors de conception.

Cependant, comme la structure physique de certains blocs ne peut être détaillée que dans des phases très tardives, cela engendre des pertes de temps et nécessite l'emploi d'outils informatiques spécifiques de modélisation des systèmes. Les démarches d'ingénierie systèmes pour le développement des produits exploitent couramment cette approche.

L'inconvénient de cette approche séquentielle est que chaque séquence verrouille certains aspects de la conception devenant des contraintes supplémentaires à la séquence suivante. De plus, non seulement le système final n'est pas optimisé, mais en plus, la durée du processus de conception s'accroît puisque la tâche suivante ne peut commencer que lorsque la précédente est terminée. Face à ces obstacles, d'autres approches ont été développées.

1.2.2.2 Démarche itérative

Cette démarche est basée sur le cycle en spirale. La plus reconnue est actuellement la méthode Agile, issue de la communauté logicielle. En effet, les approches traditionnelles séquentielles laissent peu de place aux changements. Cette impossibilité de changement est d'ailleurs un des premiers motifs d'échec des projets informatiques. Avec l'approche Agile, la notion de « gestion de projet » est remise en question au profit de la « gestion de produit », de façon à raisonner sur le « produit » plutôt que sur le « projet ».

Cette approche empirique adopte un processus de développement itératif et incrémental datant de 1986. Dans le cadre d'un projet logiciel, le client liste les fonctionnalités que le produit doit réaliser. L'équipe de développement estime le coût de chaque point de la liste pour estimer le budget global. Une partie des exigences/fonctionnalités pouvant être réalisées dans un délai très court (une itération), sont alors sélectionnées. A la fin de chaque itération, un produit partiel mais utilisable peut être présenté au client. Ce dernier peut ainsi se rendre compte très tôt de l'alignement du produit réalisé sur le besoin et émettre des retours précieux pour les futures itérations.

Le client a ainsi la possibilité de réduire « le temps de mise sur le marché » s'il estime que le produit partiel peut déjà être validé pour la production. Il a également la possibilité de changer à tout moment la priorité des fonctionnalités qui n'ont pas encore été développées en retardant une fonctionnalité (dont le besoin n'est pas encore mûr) ou en ajoutant une fonctionnalité en échange du retrait d'une autre (respectant ainsi le budget et les délais (Matthews et al. 2006)). Parmi les méthodes Agile les plus utilisées, nous pouvons citer la méthode Scrum (Martin Edin Grimheden 2013; Bricogne-Cuignières 2015).

Pour conclure, cette démarche propose une grande flexibilité dans la conception des systèmes, mais elle nécessite en contrepartie de prendre en compte au plus tôt les possibilités d'évolution du produit pour permettre de développer facilement les fonctionnalités au fur et à mesure des changements.

1.2.2.3 Approches IS/ MBSE et langage SysML

a) De l'ingénierie système à l'approche MBSE

La pratique de l'Ingénierie Système (IS) a été introduite à la fin des années 1950 pour guider l'effort technique des projets, en améliorant la communication entre les membres des équipes, tout en s'assurant de la cohérence dans la conception pour une meilleure qualité globale du produit.

La mise en œuvre de ce processus collaboratif et interdisciplinaire de résolution de problèmes, s'appuie sur les connaissances, les méthodes et les techniques issues des

sciences et de l'expérience, de sorte à définir un système qui satisfasse un besoin identifié acceptable pour l'environnement. L'objectif étant de rechercher une solution répondant à un équilibre global de tous les aspects du problème dans toutes les phases du développement et de vie du système (Meinadier 1998).

L'ingénierie système est depuis une vingtaine d'années l'objet de nombreuses recherches scientifiques (DoD 2001; National Aeronautics and Space Administration 2007; Krill et al. 2015), des communautés scientifiques se sont ainsi développées : INCOSE, ASME, IEEE autour de ce thème^{5,6,7}, et plus récemment de nombreux projets de recherche académiques et certification (PLACIS⁸, ASME Certified Systems engineers⁹) émergent autour de ce paradigme.

Par ailleurs, de nombreuses normes s'appuient également sur l'ingénierie système : la norme ANSI/EIA 632 (ANSI/EIA 2003) qui a été développée par la ECIA (Electronic Components Industry Association)¹⁰ et adoptée en 1999 par l'ANSI (American National Standards Institute)¹¹ décrit le processus d'ingénierie système, processus ensuite adapté par l'IEEE aux systèmes électriques et électroniques sous la forme du standard IEEE 1220 (IEEE 1999a).

Enfin, le standard ISO 15288 a été établi pour couvrir toutes les phases du cycle de vie du produit, par la suite standardisé par l'IEC (International Electrotechnical Commission)¹² puis par l'IEEE (ISO 2001), et enfin accompagné par la norme ISO/IEC TR 24748-2:2011 (ISO 2011a) pour guider son application.

Une des dernières tendances dans le domaine de l'ingénierie système est d'utiliser des modèles pour capturer et contrôler les informations des systèmes. Cette nouvelle approche appelée Model-Based Systems Engineering (MBSE), correspond à « l'ingénierie des systèmes basée sur les modèles ». Le fait de rassembler toutes les propriétés du système et ses informations dans un modèle unique permet de renforcer la cohérence et d'améliorer la qualité des activités de l'ingénierie système et donc du produit final. L'approche MBSE permet donc de générer des modèles au niveau système offrant la possibilité de créer une base de connaissances dans toutes les disciplines et donc de favoriser la collaboration, ainsi que la communication entre les disciplines d'ingénierie à un niveau limité de détails et donc à un niveau d'abstraction plus élevé (Hehenberger 2012). Dès lors, l'ingénierie système basée sur les modèles (MBSE) est l'application formalisée de la modélisation permettant de supporter toutes les activités de conception, de développement et des autres phases du cycle de vie (Friedenthal et al. 2008). Diverses méthodologies existent et Jeff Estefan a publié un recensement sur les principales méthodologies MBSE (Estefan 2008).

Des approches basées sur la modélisation fonctionnelle (Summers et al. 2001), très utilisées par les architectes systèmes (FBS, EFFBD (Aizier et al. 2012), SADT, Bloc Système, FPPT (Klein Meyer et al. 2007)...) proposent une décomposition successive hiérarchique des blocs,

⁵ <http://www.incose.org/>

⁶ https://community.asme.org/cie_systems_engineering_information_management_seikm/b/weblog/default.aspx

⁷ <https://www.ieee.org/membership-catalog/productdetail/showProductDetailPage.html?product=CMYSE759>

⁸ <http://www.supmeca.fr/fr/international/placis-plateforme-collaborative-d-ingenierie-systemes>

⁹ <https://www.asme.org/career-education/articles/certification/certified-systems-engineers-are-in-high-demand>

¹⁰ <http://www.ecianow.org/>

¹¹ <http://www.ansi.org/>

¹² <http://www.iec.ch/>

permettant de définir une architecture fonctionnelle avec les liens entre les différentes fonctions, et/ou les allocations physiques correspondantes (Summers et al. 2013; Ben Hamida et al. 2015a).

D'autres approches orientées objets ont également été développées en support à la démarche MBSE, tel que SysML.

b) Le langage SysML

Un autre élément important de l'approche MBSE concerne le langage utilisé. Le langage SysML (Systems Modeling Language), basé sur le langage UML 2.0 initié par l'OMG (Object Management Group) et l'INCOSE en 2001 est souvent vu par la communauté de l'ingénierie système comme le plus adapté.

Ce langage de notation graphique a été normalisé. Principalement dédié à la modélisation des systèmes pour l'ingénierie système, il vise à suppléer les nombreux tableurs utilisés dans l'industrie, dont la cohérence et la mise à jour sont souvent difficiles à maintenir. En effet, ce langage permet de spécifier, d'analyser la structure et le fonctionnement du système, de le décrire et de concevoir les sous-systèmes le composant. En outre, il permet enfin de vérifier et de valider la faisabilité d'un système avant sa réalisation, tout en offrant un modèle unique au niveau système. Dans ce contexte, chaque concepteur peut ainsi se référer à un modèle unique tout au long du cycle de conception, et dispose d'une mise à jour automatique (puisque le modèle est unique) des données partagées quel que soit l'acteur.

Ce langage a l'avantage de fournir des constructions simples et puissantes pour modéliser une large gamme de problèmes d'ingénierie systèmes. Il est particulièrement efficace pour la modélisation des exigences, de la structure, du comportement, des allocations entre les différents niveaux de modélisation et des contraintes du système et la vérification (Matthias Güdemann 2010), même s'il est souvent critiqué pour un usage industriel, à cause de la difficulté des outils commerciaux actuels à modéliser des systèmes complexes, et surtout du peu de méthodologies reconnues pour faciliter son implémentation.

Le langage SysML définit 9 diagrammes : le diagramme d'activités, le diagramme de séquence, le diagramme de machines d'état, le diagramme des cas d'utilisation, le diagramme des exigences, le diagramme de définition des blocs, le diagramme de bloc interne, le diagramme de paquetage et le diagramme paramétrique. Ces diagrammes peuvent être classés suivant leur type : les diagrammes comportementaux, le diagramme d'exigence et les diagrammes de structure (Figure 1-9). Ces diagrammes permettent une modélisation multi-vue avec différentes représentations du système reliées entre elles, pour assurer la traçabilité des éléments de modélisation entre eux.

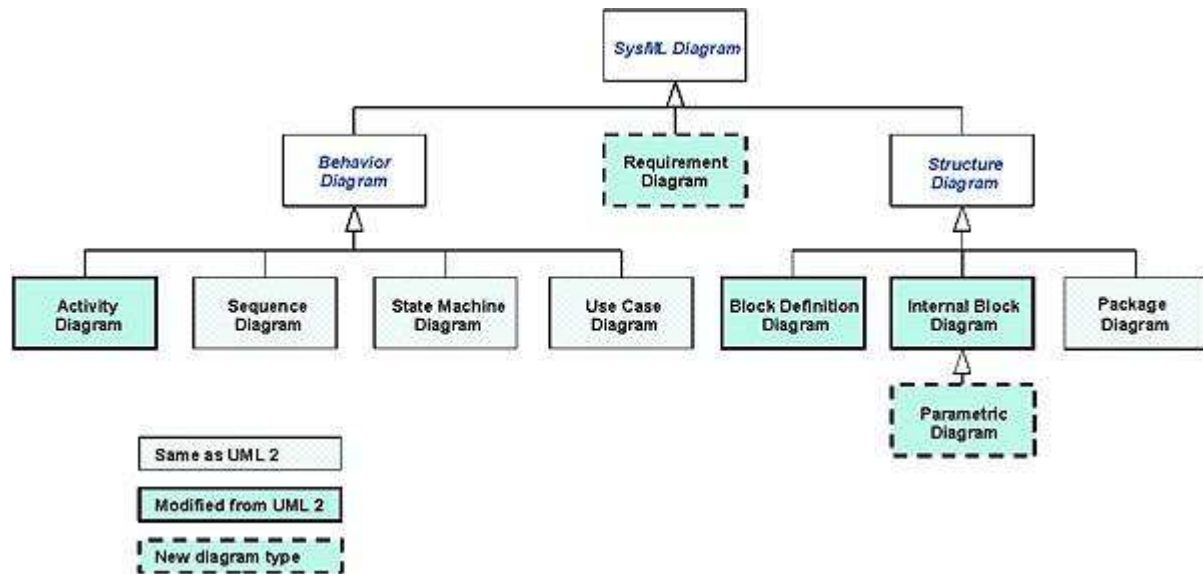


Figure 1-9 : Les différents diagrammes SysML¹³

1.2.3 Les différents points de vue de la modélisation durant la conception

Quel que soit le cycle de conception suivi, la modélisation du système va évoluer au fur et à mesure, faisant apparaître des niveaux et des types de modélisation différents.

L'approche MBSE fait apparaître deux besoins de niveaux de modélisation qui peuvent être déclinés soit au travers de leur niveau de raffinement de détails, et sont donc plutôt relatifs à la chronologie de leur apparition dans le cycle de conception, soit au travers de la nature de l'architecture générée qui sera le point d'entrée ou de sortie des équipes-métiers en charge de la conception.

1.2.3.1 Modélisation des niveaux de conception

La modélisation des niveaux de conception correspond au point de vue du projet de conception (système pour faire), qui permet d'insister sur les besoins de modélisation selon le phasage chronologique des travaux de conception (Blanchard and Fabrycky 2010).

Généralement, ces modélisations font apparaître deux phases de conception. Il s'agit des phases de « Concept » et de « Développement » (Pahl et al. 2007a).

La phase de **préconception (« Concept »)**, commence par un modèle d'exigences, suivie des modèles résultant de l'analyse fonctionnelle **spécifiant une solution de principe**. Comme modèle résultant, nous pouvons citer le modèle *d'architecture fonctionnelle* du système, et les modèles logiques.

Au début de cette phase, les exigences doivent être communes à toutes les études de développement du système : les concepts sont alors représentés sans prendre en compte leurs « supports physiques » (DoD 2001). La conception « logique », fournit ainsi le modèle *d'architecture «logique»*, où les fonctions sont traduites en éléments de comportement logique (équations, graphes d'état...). Le système est alors modélisé comme une boîte noire transformatrice de flux entrants en flux sortants. Les spécifications clients sont transformées en spécifications techniques par une analyse fonctionnelle. Les modèles utilisés dans cette phase constituent des modèles de ce que doit faire le système (Meinadier 1998; Pahl et al. 2007a). Puis cette phase de préconception intègre une phase de **conception préliminaire** qui

¹³ <http://omgsysml.org/>

va permettre d'évaluer ces différentes architectures pour retenir celle qui correspond le plus aux spécifications avant de se lancer dans des analyses plus détaillées habituellement très coûteuses (Hammadi 2012a). Ces évaluations s'appuient généralement sur des outils de simulation ou des tableurs personnalisés de chaque concepteur, permettant de faire le pré-dimensionnement des composants logiques, afin de spécifier les technologies/organes ou solutions physiques de *l'architecture organique ou physique* retenue. En effet, la phase de conception préliminaire a pour but de fournir un support permettant la prise de décision sur la **faisabilité d'une architecture système donnée** avant de passer à la conception détaillée. Les résultats de cette étape sont des modèles de simulation des différentes architectures avec une quantification préliminaire des paramètres de conception critiques pour faire le choix de l'architecture retenue. C'est à ce stade (au plus tard) que la viabilité financière du projet doit être évaluée.

Enfin, une fois l'architecture du système choisie, la phase de **développement (« Development »)** débute par la phase de *conception détaillée* (« detailed design »). Elle permet de **raffiner les paramètres géométriques et physiques** (performances, interactions physiques émergentes : thermiques, CEM...) des composants. La conception de niveau détaillé est matérialisée par des modèles géométriques 2D ou 3D des composants physiques. Le comportement physique est alors défini par des modèles mathématiques fins et des simulations bien souvent en éléments finis.

La conception détaillée définit tous les supports nécessaires à la phase de production du système. Nous pouvons citer : le placement, les formes, les dimensions et le choix des matériaux de tous les constituants, l'évaluation des différentes possibilités de production, les coûts estimés, et tous les dessins de définitions ainsi que les autres documents de production. Dès lors, la phase de conception détaillée conduit à la spécification des informations sous la forme d'un document de production (Pahl et al. 2007a).

Finalement, la phase de développement consiste à **spécifier de manière détaillée chaque pièce ou module et à définir ses dossiers d'essais ou tests unitaires pour la validation avant la production.**

1.2.3.2 Les modèles d'architectures

Ce point de vue se focalise sur le produit attendu (le système à faire), en présentant et en définissant les différents concepts d'architectures qui doivent être modélisés.

Généralement, la phase de préconception est gérée par un architecte système et son équipe, en charge de fournir un **modèle « Système »** représentant « la boîte noire » de celui-ci. Ce modèle au niveau système comprend la décomposition de l'ensemble des **architectures (fonctionnelle, logique, organique et physique)** et les allocations relatives entre elles, ainsi que les liens de traçabilité avec les exigences correspondantes.

Ainsi, nous pouvons représenter ces différentes architectures comme résultant ou alimentant les niveaux de modélisation vus précédemment sur la Figure 1-10.

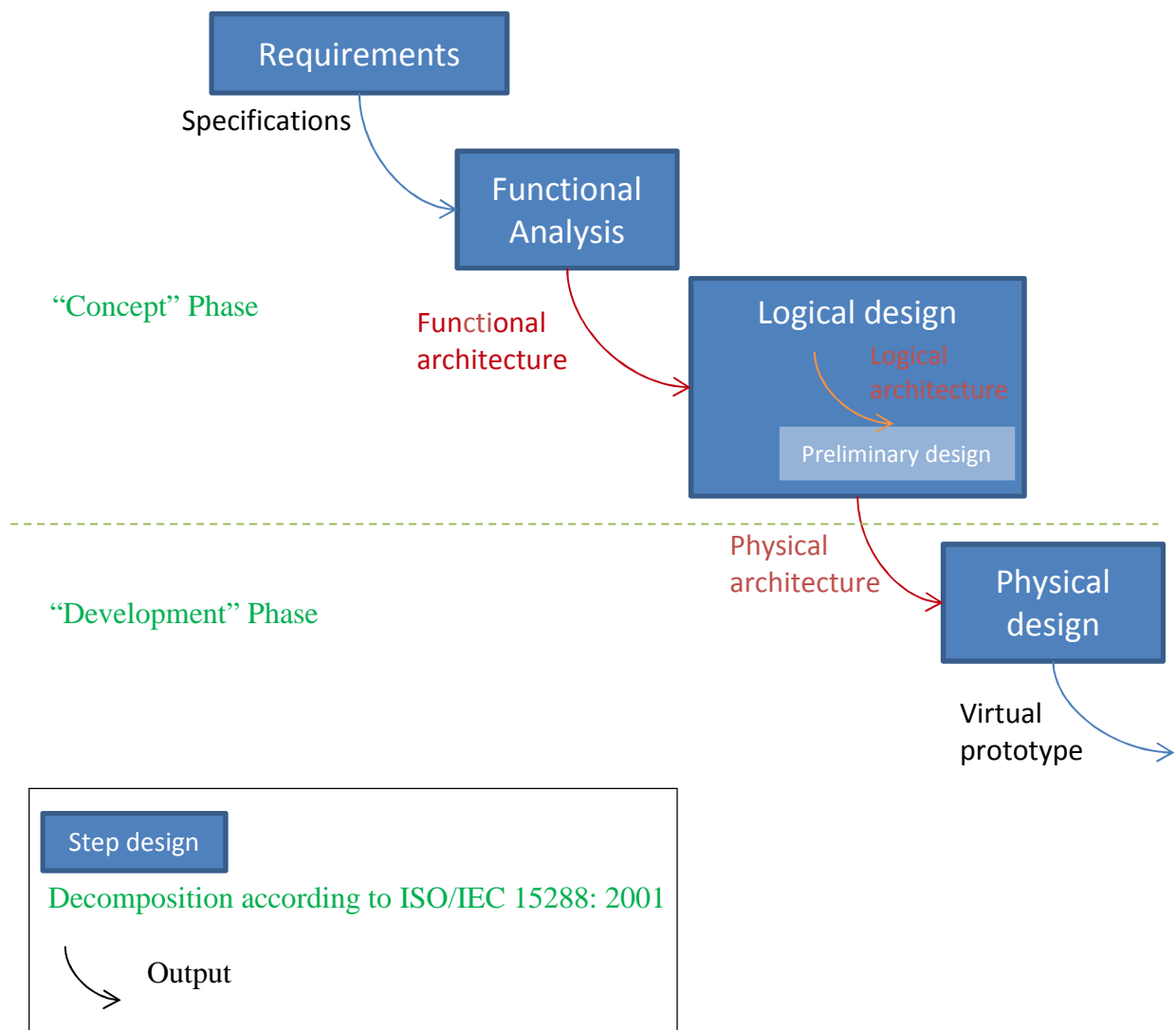


Figure 1-10 : Niveaux de modélisation de la conception et architectures associées

Voici quelques éléments de définition de ces architectures, dont le cycle de vie sera plus longuement détaillé dans le chapitre 2.

L'architecture fonctionnelle représente une décomposition fonctionnelle de type « top-down » itérative du système en traduisant les exigences de performance en fonctions et sous-fonctions. En effet, dans la vision fonctionnelle, le système est vu comme un ensemble de fonctions décomposables en deux modes de visions :

- La vision fonctionnelle externe où les fonctions sont dites de service rendues à son environnement. C'est la vision « utilisateur final ».
- La vision fonctionnelle interne où le système est vu de l'intérieur comme un ensemble de modules fonctionnels dont l'agencement restitue les fonctions de services : il s'agit de l'architecture fonctionnelle. C'est la **vision du concepteur sous l'angle fonctionnel**. Elle apporte une solution aux besoins de l'utilisateur (Meinadier 1998).

Puis, l'architecture logique met en évidence les lois physiques par des flux transformés sous-jacents (énergie brute, flux physiques...) et des flux de contrôle (informations de déclenchement et fonctions de régulation), qui nécessitent une analyse des processus dans

tous les modes de fonctionnement. Elle correspond à une **structure organisée de composants théoriques** (sans que les organes soient spécifiés).

Ces deux architectures représentent le système en termes de fonctionnalités et son fonctionnement, elles sont indépendantes des composants. Dans le processus d'IS, les architectures fonctionnelles et logiques sont confondues, et correspondent à la décomposition et à la réorganisation des fonctions résultant de l'analyse fonctionnelle, par allocation des exigences à des sous-fonctions, mettant en évidence leurs interactions. L'architecture logique correspond ainsi au dernier niveau de raffinement de l'architecture fonctionnelle, puisque le processus de conception est continuellement itératif (Hammadi 2012a).

Et enfin l'architecture physique consiste à allouer des technologies/des composants, en traçant les exigences et en définissant les interfaces : les fonctions d'interaction sont projetées sur les différents composants, mais sans définition détaillée 3D. Dans la vision physique, les organes réels de l'architecture logique sont définis et choisis : soit parmi des composants existants (composants sur étagère ou modules déjà conçus) ou à défaut, la constitution physique du système est spécifiée (spécifications techniques des besoins des constituants, matériels et logiciels restant à réaliser). Le résultat de cette approche est l'architecture physique du système qui est généralement superposable à l'architecture organique à quelques détails près. Finalement, l'architecture physique est la **solution matérielle donnée au problème initial**.

Il est possible de relever parfois, la notion d'architecture organique. Elle correspond généralement au premier niveau de raffinement de l'architecture physique définie dans l'IS, et est obtenue en projetant les et sous-fonctions sur les **organes technologiques** correspondants, pour en définir les interfaces. Les organes **réalisent les fonctions** (Meinadier 1998). L'analyse organique consiste donc à décomposer itérativement le système en constituant (sous-système, constituants, sous-constituants) jusqu'à ses constituants élémentaires (organes matériels ou modules logiciels). Le système est alors vu comme un agencement d'organes interfacés, répondant aux exigences de l'architecture fonctionnelle : c'est l'architecture organique ou technique du système.

Enfin, pour la conception mécatronique particulièrement, la continuité doit également être assurée au niveau du processus de conception que ce soit dans les différents niveaux d'architecture, modèles développés et outils utilisés, ou encore entre les équipes multidisciplinaires (Hammadi 2012a).

2 L'intégration dans la conception mécatronique

Les systèmes mécatroniques sont des systèmes multidisciplinaires qui intègrent des sous-systèmes ou des composants de différentes disciplines interagissant ensemble. Ils sont devenus de plus en plus répandus en raison de leurs nombreuses fonctionnalités et de leur fort niveau d'intégration (Bishop 2007). En outre, la mécatronique touche des applications dans divers domaines tels que le transport, la production et l'industrie médicale. La conception de ces systèmes s'appuie sur des exigences ou des fonctionnalités qui seront réalisées grâce aux interactions entre les composants. La conception traditionnelle par juxtaposition des disciplines n'est plus pertinente pour la conception mécatronique puisque

les disciplines sont de fait interdépendantes. Dès lors, la conception concourante des systèmes mécatroniques devient complexe et difficile.

2.1 Intégration et complexité des systèmes mécatroniques

La mécatronique est l'intégration synergique de la mécanique, de l'électronique, de l'automatique et de l'informatique dès le début des activités de conception pour concevoir des systèmes automatisés performants et permettre de contrôler des systèmes hybrides de plus en plus complexes (Isermann 2007). Les avantages des systèmes mécatroniques sont nombreux : ajout de fonctionnalités, réduction de la taille des composants, de l'encombrement et du poids du système global, mais aussi amélioration de la fiabilité, facilité de personnalisation des produits par reprogrammation de microcontrôleurs ou de microprocesseurs, et souvent accompagnée d'une réduction des coûts de maintenance.

En règle générale, la complexité de ces systèmes est occasionnée par l'augmentation du nombre de composants à intégrer dans un volume de plus en plus compact. Par ailleurs, ces composants intègrent de nombreuses fonctions, qui ajoutent un niveau supplémentaire de complexité. On peut notamment citer trois autres aspects de cette complexité, décrits par Chen et al. (Chen Zheng 2014). Le premier aspect de cette complexité vient du fait que les entreprises ont de plus en plus de données et d'informations à gérer, issues de différents domaines principalement de la conception, de la production et de la maintenance (Alvarez Cabrera et al. 2010). Le deuxième aspect concerne le fait que ces produits mécatroniques naissent souvent de différents experts, ce qui nécessite une collaboration soutenue entre des ingénieurs mécaniques, des ingénieurs électroniques/électriques et des ingénieurs de logiciels. En effet, les collaborateurs issus de ces différentes disciplines doivent travailler simultanément sur un modèle unique de niveau système pour comprendre l'ensemble du comportement de celui-ci et surtout les dépendances correspondantes entre leurs différentes disciplines. Le dernier aspect est que, parfois les ingénieurs sont délocalisés dans le monde et ont besoin d'outils en plus pour gérer cette collaboration.

Concernant la complexité directement liée à l'intégration mécatronique, trois types d'intégration peuvent être identifiés :

- l'intégration fonctionnelle,
- l'intégration multi-domaine,
- l'intégration physique.

Seules l'intégration fonctionnelle et physique sont explicitement définies dans la norme NF E01-010 (2008-11) (NF 2008).

L'intégration fonctionnelle consiste à intégrer le plus possible de fonctions dans un produit, soit par combinaison de plusieurs fonctions dans un seul composant, soit en dématérialisant certaines fonctions par l'utilisation de technologies informatiques programmables (par exemple, les microcontrôleurs ou les micro-processeurs). Cette « intégration fonctionnelle transversale » est souvent appelée « cross-functional integration » (Adamsson 2005) (Penas et al. 2009). Nous pouvons citer à titre d'exemple, le capteur mécanique bilame du disjoncteur Integral 32 (Schneider Electric) permettant de mesurer un courant excessif et de déclencher mécaniquement un contact. Celui-ci a été remplacé, dans le nouveau disjoncteur Tesys U, par un circuit logique programmable (ASIC) intégrant des fonctions supplémentaires telles que la transmission des informations concernant la nature des défauts, une précision améliorée, une gamme de courant élargie, l'auto-vérification du système et un convertisseur de tension,... au sein d'un même composant (Figure 1-11).

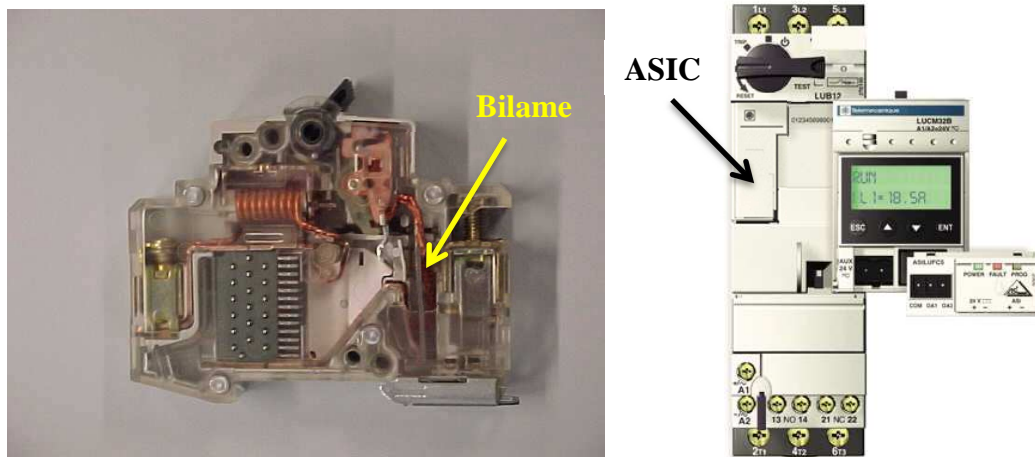


Figure 1-11 : Exemple de l'évolution mécatronique d'un disjoncteur - Integral 32 et Tesys U (Schneider Electric¹⁴)

La conception des systèmes mécatroniques met également en jeu l'intégration de plusieurs disciplines dans une même solution technologique. Cela requiert l'intégration dans un modèle unique de toutes les lois physiques, quel que soit le domaine. On parle alors d'*intégration multi-domaine* (traitée en détail dans le Chap. 2).

L'alternateur-démarrateur intégré i-Stars (Figure 1-12) est une bonne illustration de la façon dont les parties mécaniques deviennent le support des composants électroniques et comment l'électronique embarquée est distribuée dans le support mécanique.

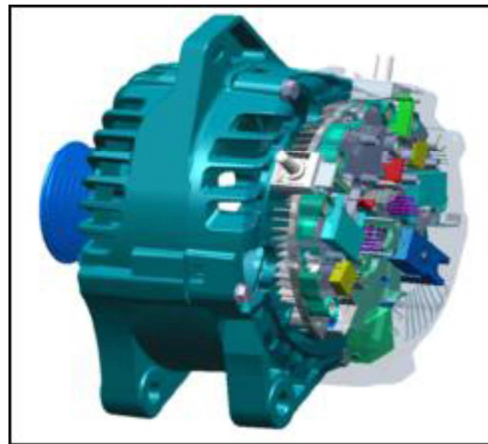


Figure 1-12 : Intégration multi-domaine : l'alternateur-démarrateur i-Stars (Valeo®)

L'intégration fonctionnelle et multi-domaine ont souvent le même objectif final qui consiste à réduire la masse et à augmenter la compacité du produit. Le système moteur-roue en est un bon exemple. En effet, celui-ci intègre une motorisation dans le volume réduit de la roue (chapitre 2 §4.1). Cette *intégration physique* de l'électronique dans des matériaux de structure est pourtant parfois très délicate.

Force est de constater que les concepteurs doivent faire évoluer leurs habitudes et travailler étroitement avec les autres services disciplinaires pour répondre aux contraintes géométriques souvent difficiles à satisfaire (O2M 2009a). Ce travail est d'autant plus ardu que la promiscuité des composants fait apparaître de nouveaux couplages multi-physiques. Cette intégration sera plus amplement détaillée au chapitre 3.

¹⁴ <http://www.schneider-electric.com/site/home/index.cfm/fr/>

2.2 Spécificités de la conception mécatronique et méthode associée

La complexité de la conception des systèmes mécatroniques est finalement basée sur des contraintes liées à la nature hétéroclite des ressources, qu'elles soient humaines ou techniques. Les systèmes mécatroniques font intervenir différentes compétences dont la coopération reste délicate à maîtriser.

Aussi, la participation des équipes de conception multidisciplinaires est nécessaire dès le début du processus de conception. Chaque module/sous-système est réalisé par des équipes différentes, avec des outils différents ne garantissant pas la cohérence entre les différents niveaux de conception. Cette particularité implique que les équipes techniques doivent travailler ensemble, dès lors les aspects collaboratifs ont une très grande importance.

Concernant les aspects multi-domaines, ils interviennent tout au long des différents niveaux du cycle de conception et peuvent causer des problèmes d'incohérences entre domaines notamment concernant le suivi et la mise à jour des données communes entre les modèles.

Par ailleurs, et même si un nombre significatif d'outils est proposé pour la conception multi-domaine des systèmes mécatroniques (Dymola, Comsol, Ansys, Catia v6, Simulink, Simplorer,...), aujourd'hui ceux-ci n'intègrent pas toutes les étapes de la conception dans un même outil (Boucher and Houlihan 2008) (Alvarez Cabrera et al. 2010), aussi il est difficile de collecter toutes les informations nécessaires, pour valider les choix de conception.

Tenant compte de ces spécificités, la complexité croissante des systèmes mécatroniques rend son processus de conception et d'aide à la décision plus difficile.

Parallèlement, peu de méthodologies sont définies pour guider le concepteur mécatronique dans sa tâche : il est de sa responsabilité d'utiliser les langages et outils les plus adaptés, en fonction de ses compétences, de sa propre expérience, de la stratégie d'entreprise (notamment pour le choix des outils), sans vision claire d'un processus de conception avec partage des données et méthodes au sein d'un processus de collaboration (Blessing et al. 1998). Finalement, une bonne méthode de conception mécatronique est celle qui offre une meilleure collaboration entre les concepteurs multidisciplinaires et qui favorise la communication entre eux, afin de maîtriser au mieux les problèmes d'interfaces entre les sous-systèmes (Hammadi et al. 2014a), tout en détectant les erreurs de conception au plus tôt. A juste titre, Hammadi (Hammadi 2012a) explique que la démarche classique de conception multidisciplinaire (pour la mécatronique) qui consiste à placer, de manière séquentielle, la structure mécanique pour lui ajouter ensuite la partie commande n'est plus adaptée pour assurer la synergie entre les disciplines de la mécatronique.

Pour faire face à ce besoin, de nouvelles approches, normes et nouveaux outils sont apparus, et notamment autour de l'ingénierie systèmes (cf. § 1.2.2.3). En effet, l'approche multidisciplinaire MBSE, basée sur les modèles, en opposition avec la méthode plus traditionnelle basée sur les documents est déjà une pratique courante dans la conception électronique ou mécanique. Sans conteste, l'ingénierie mécanique a évolué de la planche à dessin vers des outils de modélisation en 2D puis 3D dans le début des années 80. De même, l'ingénierie électronique s'appuie sur des outils de modélisation de schémas électriques et/ou de routage et de simulation de circuit (Friedenthal et al. 2008). Ainsi, lorsque ces modèles sont couplés à une démarche d'IS au travers d'une approche MBSE, celle-ci permet d'obtenir des modèles cohérents, porteurs de connaissances sur le système, tout en améliorant la communication entre les équipes de spécification et de développement, et la qualité de conception. Elle permet ainsi la réalisation de systèmes réussis, en mettant

davantage l'accent sur le système dans son ensemble plutôt que sur les différentes disciplines de manière indépendante.

Toutefois, l'existence de normes et d'outils implémentant cette approche ne suffit pas à garantir la réussite du développement de systèmes complexes. Seule une méthodologie appropriée peut conduire à une utilisation optimale des outils disponibles en fournissant un ensemble de bonnes pratiques, de procédures et de règles à suivre pendant la phase de conception.

Parmi les méthodologies existantes (méthode CORE, la méthode Harmony-SE, la méthode OOSEM (Verries 2010)...) s'appuyant sur une approche MBSE et permettant l'analyse fonctionnelle, la formalisation et traçabilité des exigences jusqu'à l'émergence de l'ensemble des différentes architectures, nous pouvons citer les travaux de Mhenni et al. (Mhenni et al. 2014). Mhenni et al. proposent une méthodologie implémentée avec le langage SysML, particulièrement bien adaptée aux systèmes mécatroniques. En effet, l'utilisation d'un langage basé sur l'ingénierie système, comme SysML, permet de générer un modèle Système, unique, prenant en considération tous les domaines des systèmes mécatroniques et ce à toutes ses phases du cycle de vie, et permet de réduire ainsi leur complexité en fournissant les mêmes paramètres à toutes les équipes techniques, pour faciliter la gestion des couplages multi-domaines (Penas et al. 2009).

Cette méthode basée sur une analyse « en boîte noire », analogue à l'analyse externe et une analyse « en boîte blanche » correspondant plutôt à l'analyse interne du système est décrite ci-dessous, avec une illustration autour d'un télescope GoTo.

Les étapes préalables à l'émergence de la première architecture (architecture fonctionnelle) constituent l'« analyse en boîte noire ». Celle-ci est basée successivement sur :

- un premier diagramme d'exigences (req), recueillant les principales exigences du client (Figure 1-13),

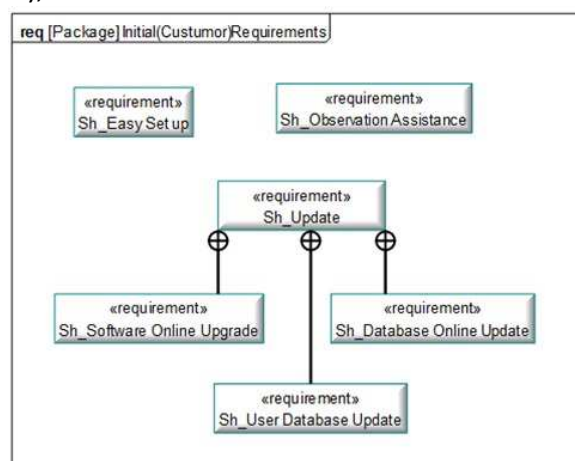


Figure 1-13 : Diagramme d'exigences (req) (Mhenni et al. 2014)

- un diagramme du cycle de vie (stm) présentant les phases du système à prendre en compte pour extraire les exigences de l'ensemble des phases du cycle de vie (Figure 1-14),

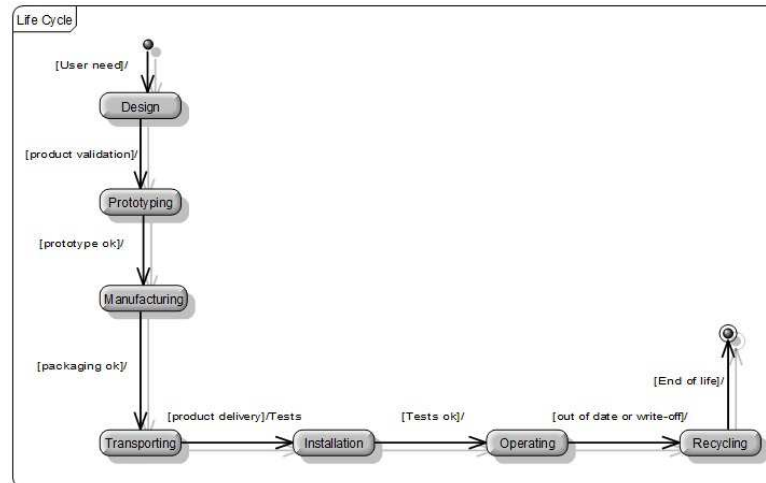


Figure 1-14 : Diagramme du cycle de vie (stm) (Mhenni et al. 2014)

- le diagramme de contexte (bdd), pour une phase de cycle de vie donnée, définissant le périmètre du système et les acteurs externes en interaction avec celui-ci (Figure 1-15),

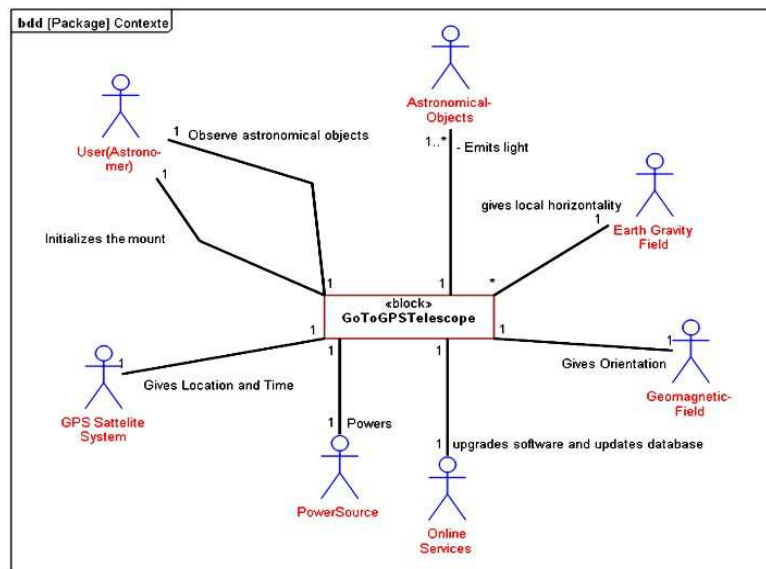


Figure 1-15 : Diagramme de contexte (bdd) (Mhenni et al. 2014)

- un diagramme d'interfaces externes (ibd) pour définir la nature des flux externes entrant et sortant, échangés avec les acteurs extérieurs définis précédemment (Figure 1-16),

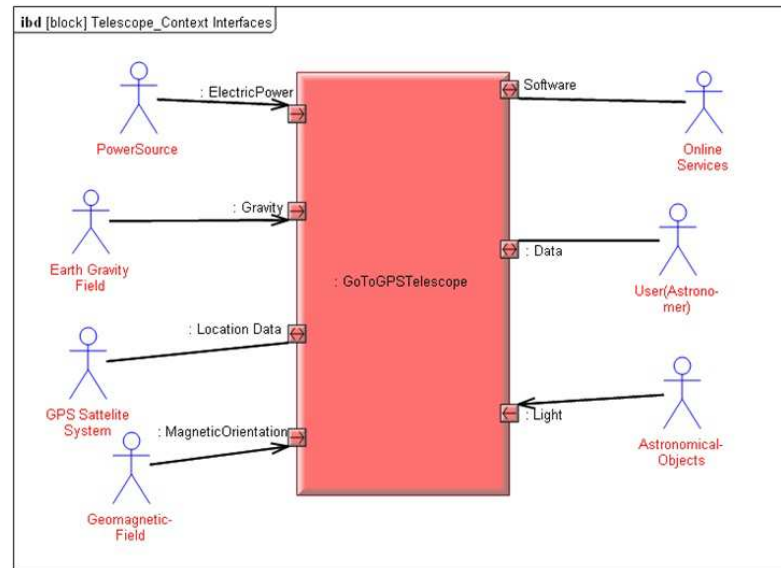


Figure 1-16 : Diagramme d'interfaces externes (ibd) (Mhenni et al. 2014)

- un diagramme de modes de fonctionnement du système (stm) (Figure 1-17),

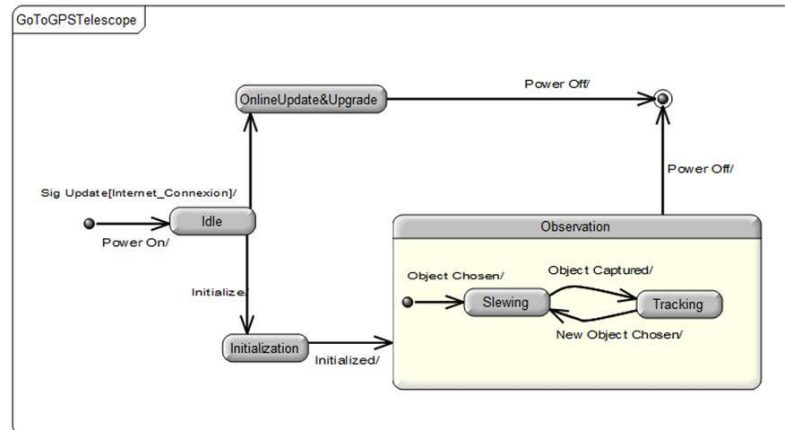


Figure 1-17 : Diagramme de modes de fonctionnements du système (stm) (Mhenni et al. 2014)

- des diagrammes de cas d'utilisation (ucd) pour formaliser les services rendus par le système dans un mode de fonctionnement donné (Figure 1-18),

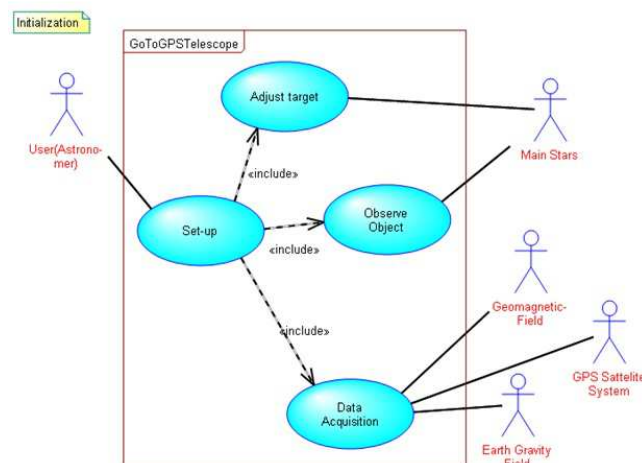


Figure 1-18 : Diagramme de cas d'utilisation (ucd) (Mhenni et al. 2014)

- et enfin un diagramme de séquence (seq) permettant d'établir le déroulement temporel des cas d'utilisation et ainsi définir les premières fonctions internes que le système doit remplir (Figure 1-19).

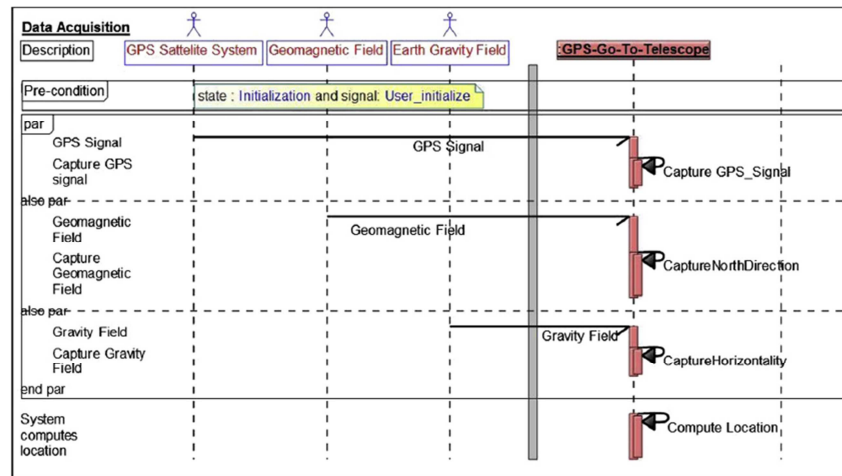


Figure 1-19 : Diagramme de séquence (seq) (Mhenni et al. 2014)

Cette première phase d'analyse externe se termine par la formalisation et la traçabilité de l'ensemble des exigences vis-à-vis des éléments du modèle précédemment décrits dans un diagramme d'exigences (Figure 1-20).

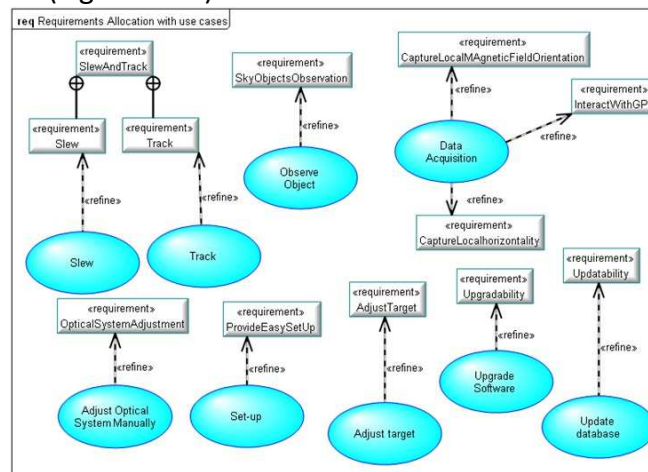


Figure 1-20 : Diagramme de traçabilité des exigences

Dans l'analyse « boîte blanche », l'architecture du système est progressivement déterminée. D'abord, un modèle hiérarchique de la répartition des fonctions du système est établi au travers de diagrammes d'activités (act) (Figure 1-21).

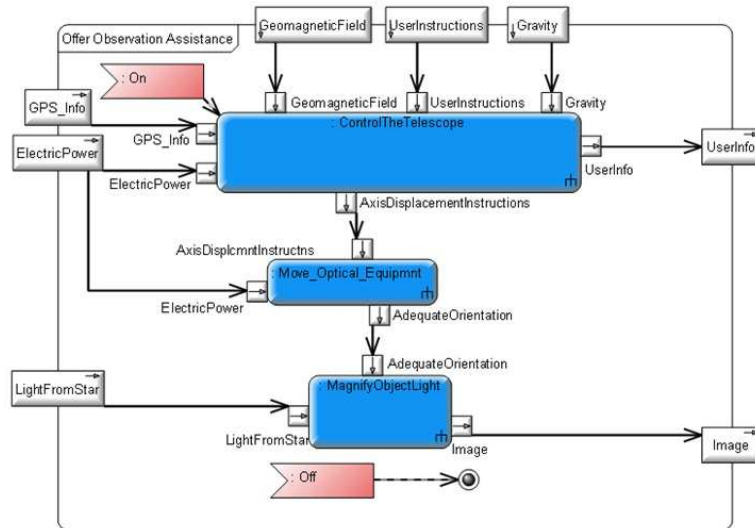


Figure 1-21 : Diagramme d'activités (act) (Mhenni et al. 2014)

Ensuite, sur la base de cette architecture fonctionnelle, le diagramme de structure de bloc « block definition diagram » (bdd), permet de définir les différentes architectures logiques candidates. Une architecture logique est basée sur des catégories générales (technologies) de composants répondant aux fonctions précédentes, sans spécifier les caractéristiques détaillées des composants (Figure 1-22). Notons que ce diagramme représente les éléments structurants appelé blocs avec leurs propriétés, leurs relations et leur composition. Il offre une vue statique des différents niveaux d'architectures, en intégrant les liens d'allocations entre les éléments des différents niveaux d'architecture.

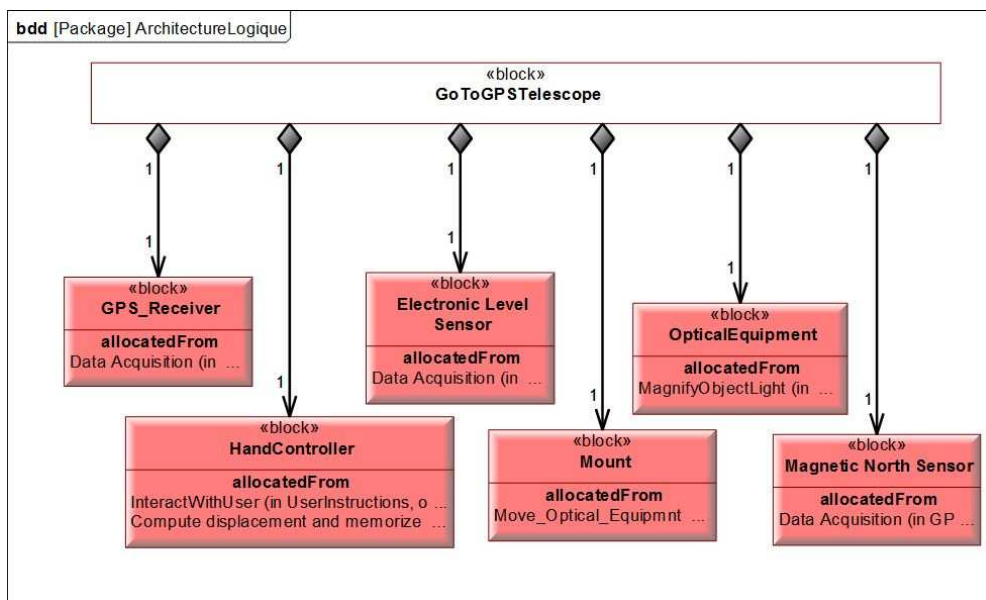


Figure 1-22 : Diagramme de définition de bloc décrivant une architecture logique (Mhenni et al. 2014)

Aussi, pour chaque architecture candidate, un diagramme interne de blocs (ibd) peut être aussi utilisé pour avoir une vision plus « dynamique » de l'architecture, en présentant les flux d'entrées/sorties et leur nature entre chaque entité (Figure 1-23).

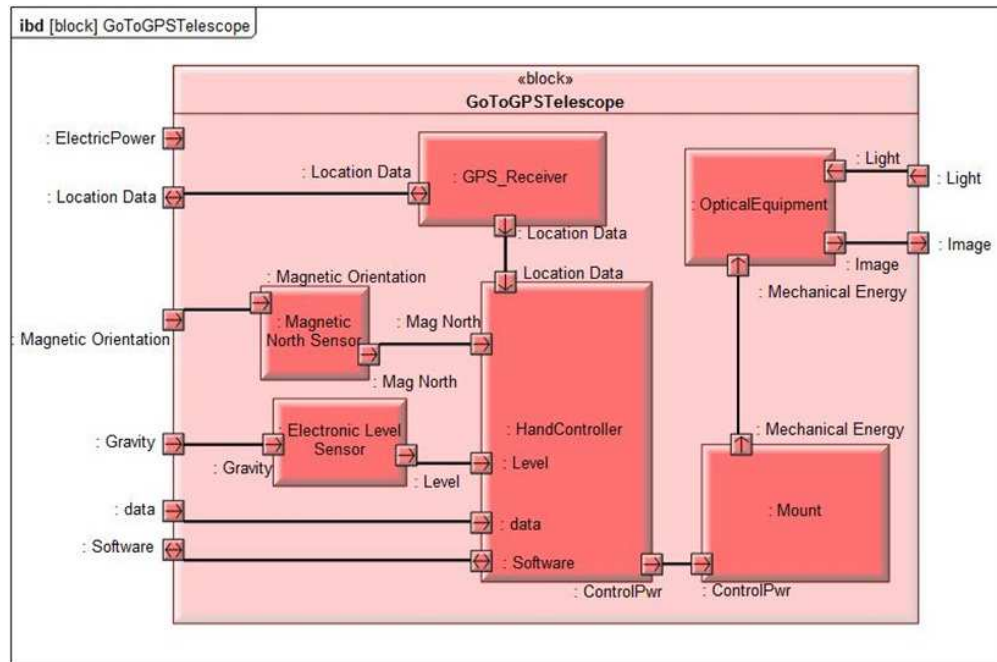


Figure 1-23 : Exemple de diagramme interne de définition de blocs pour l'architecture logique (Mhenni et al. 2014)

Enfin, des compromis permettent de définir l'architecture physique correspondant à l'affectation des composants physiques optimaux (composants sur étagères ou à concevoir) aux entités logiques précédentes (Mhenni et al. 2014). Ces choix sont soit issus de résultats de simulation de diagrammes paramétriques (par), soit basés sur des simulations complémentaires avec des outils externes tels que présentés au chapitre 2 § 2.3.4.5.

Un diagramme paramétrique relie les paramètres des composants ou du bloc système les uns aux autres par des « blocs contraintes » portant l'ensemble des équations (Figure 1-24). De manière générale, il vise à identifier les paramètres principaux du système ainsi que leurs relations au travers d'équations afin de pouvoir les exporter vers d'autres outils de simulation pour les résoudre.

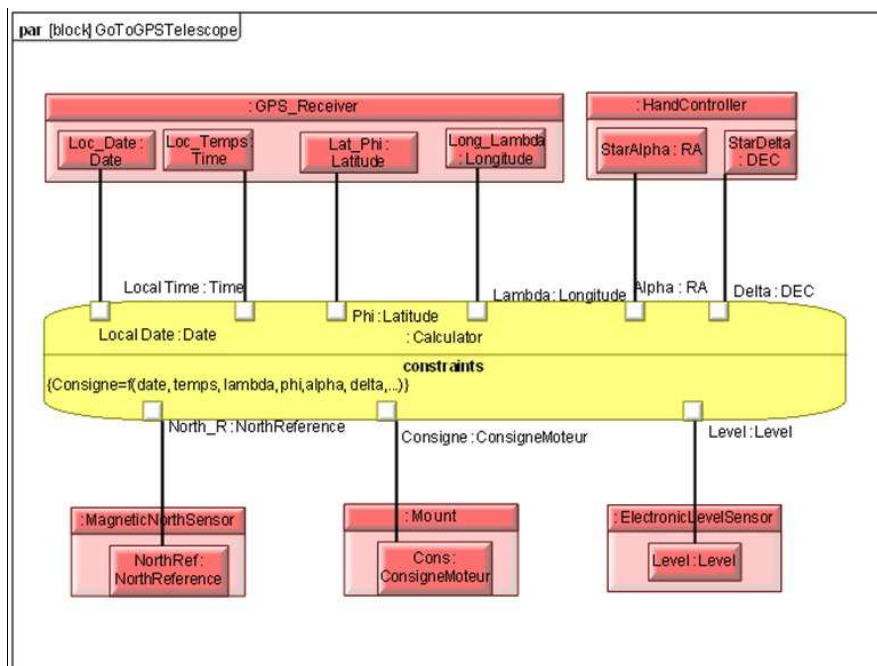


Figure 1-24 : Exemple de diagramme paramétrique (Mhenni et al. 2014)

2.3 Enjeux pour l'évaluation de la conception mécatronique

La conception des systèmes mécatroniques est particulièrement longue et difficile, du fait d'une part de leur forte intégration fonctionnelle, de leurs aspects multi-domaine et multi-physique et d'autre part des couplages qui en résultent.

A chaque étape du cycle de conception, à tous les niveaux d'abstraction ou de décomposition du système, le concepteur doit être capable d'étudier les compromis et de justifier les choix de conception à partir des analyses des différentes disciplines, en garantissant la cohérence de la solution. Il doit fournir une solution équilibrée, en étudiant les compromis successifs dans une vision globale, pour permettre les choix de conception, ceci dans un ensemble d'attentes et de contraintes de tout ordre, généralement contradictoires. S'appuyant sur un travail d'équipe pluridisciplinaire issu du contexte d'intégration précédemment décrit, le concepteur doit ainsi déterminer les ressources nécessaires, les impacts fonctionnels et de performances sur le projet (délais, coût), ainsi que les risques des différentes solutions alternatives (Meinadier 1998).

Dès lors, une interrogation apparaît : comment aider le concepteur dans ses choix pour lui permettre d'évaluer ses solutions objectivement ? Un des moyens est d'utiliser des métriques pour évaluer quantitativement à quel degré la solution candidate répond aux objectifs fixés. Cette quantification est essentielle pour pouvoir limiter les itérations en aval du cycle de conception (Turki 2008a). En effet, la plupart du temps, les concepteurs, de par la variété de leur formation, leurs compétences et leurs expériences, ont des analyses et interprétations différentes, qui ne permettent pas d'arriver rapidement à des compromis, sans la mise en œuvre de support d'aide à la décision collaboratif (Jankovic et al. 2006). A l'évidence, l'utilisation de critères mesurables pour l'évaluation de la conception et notamment le choix d'architectures (Jankovic et al. 2012) est nécessaire pour avoir des arguments objectifs (Moody 1998).

3 Aide aux choix de conception mécatronique : les métriques

3.1 Intérêt des métriques

Les métriques sont un support d'aide à la décision intéressant dans la formalisation et la traçabilité des critères sur lesquels les décisions finales seront prises. Elles permettent de capitaliser l'expérience des concepteurs, et peuvent intégrer le retour d'expériences des utilisateurs, etc. A ce titre, les métriques sont susceptibles de fournir un support pour le planning, la prévision, la surveillance du processus de conception, offrant une traçabilité du changement des exigences, la validation et la mesure de l'efficacité des revues de projets (Kasser and Schermerhorn 1994; Xijuan et al. 2003), mais également le contrôle et l'évaluation de la qualité des produits et des processus (Briand et al. 1998). Elles ont donc vocation à être utilisées pour de nombreux d'objectifs tout au long du cycle de conception.

Plus spécifiquement pendant les étapes préliminaires de la conception, où les ingénieurs et les managers ont besoin d'évaluer les projets de conception ou les concepts du système (McAdams and Wood 2002) pour pouvoir prédire leur succès (Fenstermaker et al. 2000), il est fondamental de leur fournir des moyens pour l'évaluation de ces projets, concernant notamment : la maturité du projet, le respect du planning, le degré d'innovation ou de collaboration effective et le niveau d'effort (économique, humaine,...).

Une fois la faisabilité du projet acceptée, les métriques peuvent être également utiles pour l'aide à la décision des choix de conception du système à développer : métriques pour le choix des architectures et des solutions physiques, métriques de coût ou de valeur ajoutée (Ittner et al. 2003), métriques de vérification... En effet, les métriques permettent de garder la traçabilité des choix de conception autant pour les solutions retenues que pour les solutions recalées et ainsi participent en permanence à l'amélioration de la performance des systèmes (Clivillé 2004). Par ailleurs, elles fournissent également la justification d'un modèle et par exemple, la raison pour laquelle une simulation doit être menée dans la phase de conception détaillée (Hammadi 2012a).

Un dernier intérêt des métriques peut être soulevé, et non des moindres pour les industriels : il s'agit de l'évaluation de la qualité du produit qui se détermine par la mesure de ses performances (Neely et al.), de son efficacité, de sa pertinence, de sa complexité, de sa modularité, de sa fiabilité pour ne citer que les plus récurrentes.

Pour résumer, les intérêts des métriques sont nombreux et leurs différentes utilisations tout au long de la conception seront détaillées au paragraphe 4 de ce chapitre. Soulevons maintenant leur intérêt plus particulièrement pour la phase de conception.

Une lecture de l'état de l'art au sein de la communauté d'ingénierie logicielle, montre que les métriques existent depuis les années 60, et un grand nombre d'entre elles et d'outils d'évaluation ont déjà été développés pour mesurer la complexité, la qualité et l'efficacité des codes (Ince and Sheppard 1988; Briand et al. 1994; Poels and Dedene 1999).

Jones, (Jones 2008) par exemple, a proposé de classer ces métriques en quatre catégories : les mesures de qualité, de productivité, les mesures de planning et les mesures d'entreprise. Pour la communauté logicielle, une métrique est un ensemble de mesures issues des propriétés techniques ou fonctionnelles d'un logiciel. C'est un moyen ou une méthode permettant de vérifier la qualité des modèles logiciels, en s'appuyant sur des indicateurs calculés sur ces modèles (Turki 2008a).

Ainsi, bien que des métriques se soient déjà largement développées depuis des décennies pour la communauté logicielle (Côté et al. 1988), les challenges de la conception mécatronique ont aujourd'hui créé les mêmes besoins pour cette discipline multi-domaine, et jusqu'à présent peu d'études (Hehenberger and Zeman 2005; Warniez et al. 2012) ont proposé des métriques spécifiques pour ces systèmes complexes.

En effet, actuellement, trop de processus de conception mécatronique sont uniques et dépendants trop souvent de l'expérience ou de l'intuition d'ingénieurs expérimentés, sans support méthodologique ou de validation spécifique. Le très grand nombre de données de conception communes aux différentes disciplines d'ingénierie impliquées est une problématique importante pour les concepteurs des systèmes mécatroniques. Aussi, il devient capital pour la conception de ces systèmes de fournir des informations prédictives au plus tôt dans leur cycle de vie. Par ailleurs, la conception des systèmes mécatroniques génère un espace de solutions possibles si large que les concepteurs doivent disposer de moyens pour sélectionner la solution optimale satisfaisant un grand nombre de contraintes multidisciplinaires. Enfin, la complexité croissante de ces systèmes due à l'intégration d'éléments multi-domaines connectés et interactifs rend difficile l'utilisation de métriques existantes mono-domaines. En effet, la conception des systèmes mécatroniques étant basée sur des équipes techniques différentes (Electronique, Automatique, Mécanique, Informatique), il est devenu nécessaire non seulement d'aider l'architecte système à choisir

l'architecture optimale du système, mais aussi d'évaluer en amont les besoins collaboratifs du projet, ou la fiabilité du système global.

Reste qu'en l'état, même si les métriques existent depuis longtemps, leur définition n'est à ce jour, pas encore standardisée...

3.2 Définition des métriques

Il existe dans la littérature, plusieurs définitions du concept de métrique, dont le vocabulaire diffère selon le domaine d'application, voire au sein même du domaine de la conception de systèmes.

Généralement, les auteurs s'accordent pour dire qu'une métrique permet de mesurer et donc de comparer plusieurs choses. Dans le domaine de la conception, il s'agit souvent d'évaluer quantitativement plusieurs solutions au travers de divers critères, qui seront combinés, avec éventuellement certaines pondérations pour obtenir une valeur globale représentative de ce que l'on cherche à évaluer (objectif).

Les différences observées suivant les auteurs concernent la définition de ces critères.

Si ces critères font intervenir des valeurs de référence, ils seront dénommés « indicateurs techniques » (Lo 2013) ou « de performance » (Addouche et al. 2005).

Ainsi, une métrique se définit par une combinaison (fonctions mathématiques) de critères ou indicateurs mettant en jeu un ou plusieurs paramètres directement mesurables. Ces fonctions mathématiques peuvent être simples comme la somme, la soustraction, la multiplication, la division, mais aussi les fonctions polynomiale, logarithmique, exponentielle, ou des équations plus complexes.

Les critères ou indicateurs sont établis à partir de variables ou paramètres influents préalablement identifiés.

Nous avons essayé de synthétiser ces différentes définitions et vocabulaires associés dans la Table 1-1 suivant les auteurs, en se basant sur la décomposition présentée en Figure 1-25.

La figure suivante (Figure 1-25) permet d'illustrer les relations entre les différents termes qui sont utilisés dans la littérature et repris dans la Table 1-1.

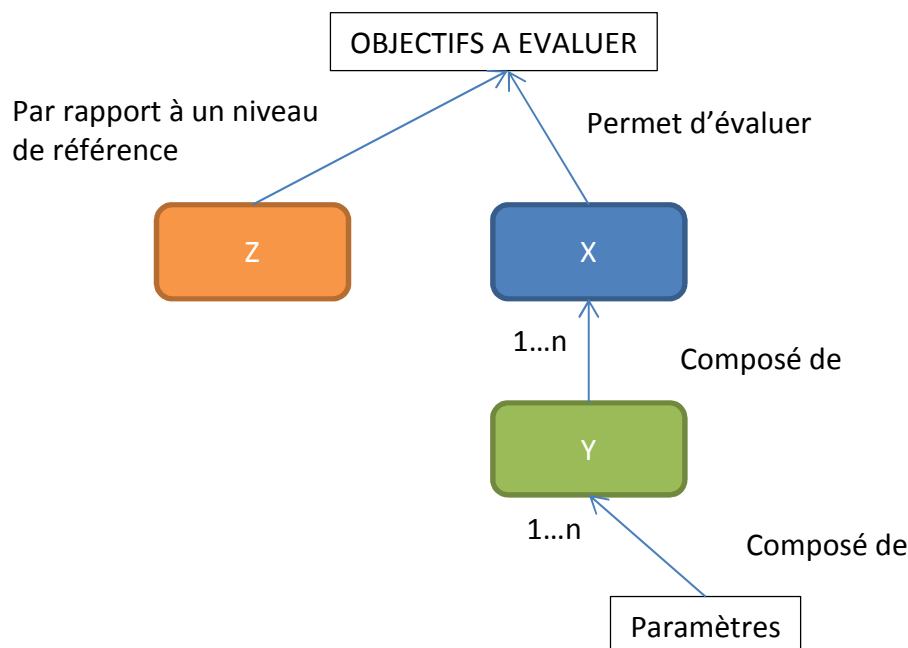


Figure 1-25 : Schématisation de la définition d'une métrique à partir de données de la littérature

Littérature	Objectif à évaluer	X	Y	Paramètres	Z
(Lo 2013)	Objectif	Critères d'évaluation	Indicateurs techniques	-	Valeur cible
(Scaravetti 2004)	Critères globaux (critères de qualification concepteur)	Indicateur de pertinence	-	Variables critères	-
(Zwingmann 2005)	Caractéristiques du produit	Indicateur	Critères	Facteurs influents/variables de conception	-
(Addouche et al. 2005)	Objectif	Mesure	Indicateurs de performances	Variables	Performance

Table 1-1: Tableau récapitulatif des termes de la définition d'une métrique

Pour Lo (Lo 2013), l'objectif à évaluer est composé de *critères d'évaluation*. Un *critère d'évaluation* est une règle de jugement sur la satisfaction d'un *objectif* spécifié. Le critère est exprimé à partir d'une ou plusieurs exigence(s) technique(s) et il est précisé par un ou plusieurs *indicateurs techniques*. Les *indicateurs techniques* précisent les *critères d'évaluation* et sont une grandeur mesurable possédant une valeur cible (correspondant à une exigence de performance) et une valeur estimée ou calculée à partir du modèle. Lo en distingue trois types :

- les mesures d'efficacité (Measure of Effectiveness : MoE),
- les mesures de performances (Measure of Performance : MoP)
- les mesures de performance techniques (Technical Performance Measure : TPM),

Pour Scaravetti (Scaravetti 2004), les *critères globaux* (ou encore critères de qualification concepteur) sont utilisés tout au long du cycle de conception et sont définis lors de l'analyse fonctionnelle sous forme de fonctions contraintes. Ce sont donc des critères d'appréciation et leur valeur exprime la pertinence d'une solution. A titre d'exemple, ils permettent d'évaluer le coût, le rendement, la sûreté de fonctionnement (fiabilité, maintenabilité, disponibilité, sécurité), l'évaluation de l'écoconception, l'évaluation du démantèlement/recyclabilité. Ces critères globaux sont en fait l'objectif à évaluer. Ils sont ensuite traduits en *indicateurs de pertinence*. Les indicateurs de pertinence sont alors définis à partir des *variables critères*.

Zwingmann (Zwingmann 2005), développe l'idée selon laquelle l'évaluation se fait à l'aide d'*indicateurs* permettant d'évaluer les caractéristiques du produit (l'objectif à évaluer) et de vérifier le respect des règles de « bon comportement ». Ces indicateurs sont composés de *critères* fixés par l'analyse fonctionnelle du produit. Ces derniers sont eux-mêmes composés de *facteurs influents* qui sont caractérisés par des *variables de conception* de bas niveau.

Enfin, (DoD 2001) donne une autre définition des *métriques* comme des mesures permettant d'évaluer la progression d'un processus (ou d'un projet), par l'observation de changements des valeurs de mesures. L'auteur précise que ces métriques sont de 3 types :

- Les métriques de performance : pour mesurer la performance du produit;
- Les métriques de pertinence : pour mesurer la sécurité, la maintenabilité, la fabricabilité, la modularité, etc. ;

- Les métriques d'accessibilité : pour mesurer les coûts.

Définition proposée

En s'inspirant des définitions existantes, nous proposons, dans la suite des travaux de la thèse, une définition des métriques faisant la synthèse de ces définitions. Nous définissons une **métrique comme l'expression mathématique permettant d'évaluer la conception d'un système selon un objectif défini en identifiant des indicateurs représentant les contraintes de la conception et caractérisés par un ou plusieurs paramètres influents. Ainsi une métrique peut être la somme de plusieurs indicateurs**. La valeur de cette métrique sera sans unité dans un intervalle borné. Chaque borne correspondra à la valeur idéale de l'objectif à atteindre ou à éviter.

Règles de définition des métriques

Afin de construire des métriques permettant une évaluation quantitative et scientifique de la conception, les métriques développées devront remplir quelques conditions extraites des travaux de Lawson (Lawson 2005) et plus surtout d'Heinrich et al. (Heinrich et al. 2007) qui ont travaillé plus spécifiquement sur les métriques évaluant la qualité des données. Les règles sélectionnées sont les suivantes :

- Une métrique doit être normée. En effet, une normalisation adéquate est nécessaire pour garantir que les valeurs des métriques soient comparables. Dans ce contexte, les métriques sont souvent un ratio avec une valeur comprise entre 0 (opposée aux objectifs du concepteur) et 1 (conforme aux objectifs du concepteur)
- Les valeurs de la métrique doivent être comprises dans une échelle d'intervalle adéquate pour supporter et anticiper les changements de niveaux des valeurs dans le temps. Cela signifie que la différence entre deux niveaux doit être significative et doit rester cohérente dans tout l'intervalle. Par exemple un écart de 0,2 entre les valeurs de 0,4 et 0,6 doit avoir la même signification qu'un écart de 0,2 entre les valeurs 0,7 et 0,9.
- La métrique doit être compréhensible par les concepteurs des différentes équipes. Elle doit être facile à interpréter.
- Une métrique doit être réalisable. Les métriques doivent être basées sur des paramètres d'entrée qui sont déterminables. Cela permettra l'automatisation de la procédure de mesure.

Construction d'une métrique

Le processus de construction d'une métrique dans un contexte industriel que nous proposons est détaillé dans le *Chapitre 4*.

4 Métriques dans le cycle de conception

Les métriques ne sont pas un sujet de recherche récent, puisqu'elles ont été initiées et largement développées par les informaticiens pour évaluer la complexité des codes, un peu par les électroniciens et les automaticiens pour des critères d'observabilité, de commandabilité, mais beaucoup moins et voire sont quasi-inexistantes dans le domaine de la mécatronique.

En effet, les méthodes d'évaluation dans la conception des systèmes mécatroniques sont souvent issues des domaines de l'électronique, de la mécanique, du logiciel et sont utilisées indépendamment, sans prendre en compte l'intégration des disciplines qui caractérise ces systèmes. Aussi, nous présentons dans ce paragraphe un état de l'art général des métriques pouvant être utilisées pendant la conception de produits, selon la phase de conception où elles interviennent. Il s'agit principalement des métriques de préconception, de conception préliminaire, de conception détaillée, et de vérification – validation (Figure 1-26)

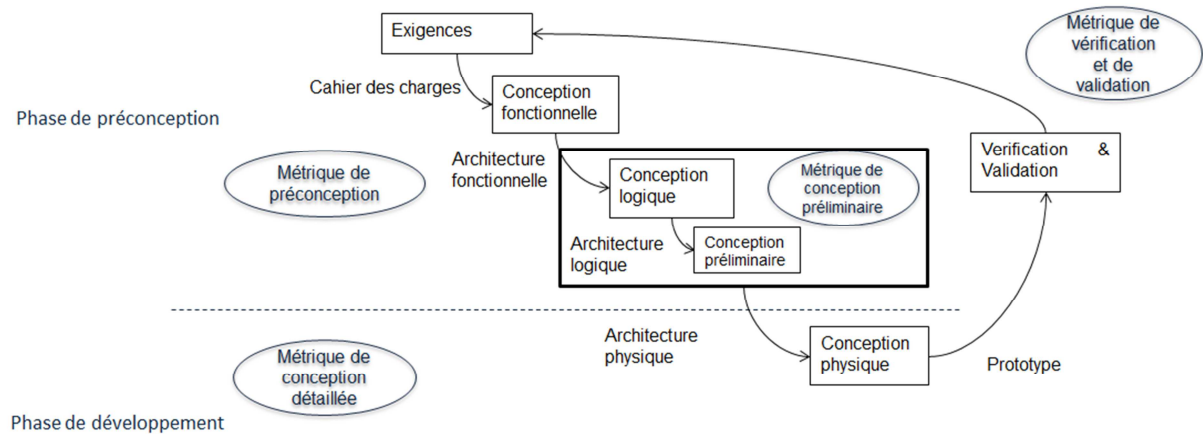


Figure 1-26 : Les métriques dans le cycle de conception

4.1 Les métriques en phase de préconception

Les métriques en phase de préconception, également appelées « métriques conceptuelles » permettent d'évaluer les différents concepts et aident au choix du concept à retenir. Elles sont basées sur l'analyse des exigences et de la sensibilité des clients à ces exigences. Les paramètres influents seront par exemple : le temps de développement et le coût prévisionnel d'un projet, le savoir-faire industriel, les compétences de chacune des équipes de conception impliquées, etc. Ainsi pour évaluer la faisabilité d'un concept, il est nécessaire de mesurer la maturité du projet, ainsi que les autres risques du projet, son coût, le planning envisagé, les performances techniques du concept, en adéquation avec les souhaits du client.

4.1.1 Les métriques de maturité

Heinrich et al. (Heinrich et al. 2007) définissent une métrique de maturité des données, pouvant aisément s'appliquer à la maturité d'un concept, en faisant la somme des estimations de la maturité des données disponibles du projet. Cette métrique, comprise entre 0 et 1 permet ainsi de calculer la maturité des données tout au long du cycle de conception :

$$M_{maturité} = e^{(-incertitude \times \text{âge de la donnée})} \quad (1.1)$$

Le paramètre *incertitude* représente l'incertitude de la validité de la donnée. Aussi, pour les valeurs qui ne changeront jamais, l'incertitude sera égale à 0 et donc la métrique sera égale à 1 étant donné que $M_{maturité} = e^{(-0 \times \text{âge de la donnée})} = 1$

Le paramètre *âge de la donnée* est calculé en fonction de l'instant où il est quantifié et le moment où il est utilisé. Si la valeur de la métrique est calculée au moment même de la

quantification de la donnée (dans le cycle de conception par exemple), l'âge de la donnée est égale à 0 et la métrique vaut également 1 puisque $M_{maturité} = e^{(-incertitude \times 0)} = 1$

Lors de la ré-acquisition de la valeur de la donnée, celle-ci est alors considérée comme une mise à jour de cette donnée existante.

4.1.2 Les métriques de risques

Le risque se définit comme la probabilité d'«un évènement incertain qui pourrait entraîner un impact incertain sur le calendrier, le coût ou la qualité d'un produit» (Ferguson 2004), (ISO 2001).

Ainsi, il est possible de mesurer le risque sur l'ensemble du cycle de vie du produit pour faire des projections sur le succès ou l'échec probable d'un projet de conception d'un système.

Les métriques de risques sont utilisées pour comprendre et éviter les risques potentiels de *coût*, de *planning* et de *performances/techniques* d'un système.

Le *risque de coût* évalue la possibilité que le budget disponible soit dépassé, que ce soit au niveau du projet total ou pour un élément particulier du système.

Le *risque de planning* évalue si le projet ne puisse pas atteindre les jalons prévus, parce que les délais sont trop courts, ou qu'il y a des difficultés prévues dans la réalisation technique. Les effets en cascade des risques de planning au niveau d'un élément peuvent produire un risque de calendrier sur l'ensemble du projet.

Le *risque technique* est la possibilité qu'une exigence technique du système ne puisse pas être atteinte dans le cycle de vie de celui-ci. Le risque technique existe donc si le système n'est pas sûr d'atteindre les exigences de performance et/ou de répondre à l'opérabilité, la productibilité, la testabilité, aux exigences d'intégration, ou encore aux exigences de protection de l'environnement (INCOSE 2006).

Le risque (R) est exprimé par le produit de la probabilité (P) qu'un évènement se produise et de la conséquence indésirable (V) d'un évènement s'il se produit :

$$R = P \times V \quad (1.2)$$

Les métriques suivantes proposées par Ferguson (Ferguson 2004) correspondent à l'«analyse qualitative des risques». Ces métriques donnent ainsi un point de vue à la direction ou aux managers pour déterminer parmi plusieurs projets celui qui a le niveau de risques le plus élevé.

La métrique du risque total pour un projet est la valeur de la somme de tous les risques :

$$Métrique_{Risques} = \sum_{i=1}^{nR} R_i \quad (1.3)$$

Avec nR le nombre total de risques.

La probabilité d'un évènement (P) est décomposée en 5 niveaux (Figure 1-27) :

Label	Description	Value
Very Low	In your career, you have never seen this happen, but it could.	5% Range 1-9%
Low	It has happened on occasion.	25% Range 10-29%
Moderate	Sometimes it happens and sometimes not.	50% Range 30-69%
High	Most of the time this event will occur.	75% Range 70-89%
Very High	It has happened on every project, and you think it always will, but there is a chance of escape.	95% Range 90-99%

Figure 1-27 : Niveaux de probabilité de risques d'un évènement (Ferguson 2004)

De la même manière, la valeur d'impact (V) est décomposée en 5 niveaux (Figure 1-28) pour une métrique normée. Le maximum est atteint lorsqu'il y a un dépassement de calendrier ou de budget ou lorsqu'il n'y a pas de client (pas de marché, pas de vente,...). Dans la pratique, un chef de projet peut gérer un risque de niveau 1 ou 2.

Impact Level	Impact Value
5	81
4	27
3	9
2	3
1	1

Figure 1-28 : Niveaux de valeur d'impact d'un risque (Ferguson 2004)

La valeur maximale de la métrique est donc de $95\% \times 81 = 76,95$ et la valeur minimale est de $5\% \times 1 = 0,05$. En général, pour calculer l'évaluation totale du risque d'un projet en préconception, il n'est pas nécessaire de calculer la somme de tous les risques pour le projet. En effet, à ce stade de la conception certains sont inconnus et l'usage est de prendre la somme des 20 plus hauts risques du projet pour obtenir un aperçu du risque total (Ferguson 2004).

4.1.3 Les métriques de coûts

Lorsqu'un choix de concept doit être fait, les décideurs doivent pouvoir vérifier que le « business plan » et le coût associé au développement du produit choisi sera acceptable, pour parvenir à un retour sur investissement (ROI : Return on Investment (Poulin and Caruso 1993)) économiquement intéressant pour l'entreprise. Ainsi, dès la phase de préconception, les métriques de coûts doivent être prises en considération pour valider le concept proposé. Les métriques de coûts durant la phase de préconception prennent en compte l'estimation des budgets et des ressources nécessaires pour le développement du produit et sa fabrication (DoD 2001).

Parmi ces métriques de coûts, les métriques de faisabilité financière fournissent également une estimation du coût de production, un suivi tout au long de la phase d'ingénierie de la conformité aux objectifs (DTC : Design-to-Cost, méthode pour identifier les problèmes de

conception liés aux coûts de production (Deepak 1998)). Toutes ces estimations sont combinées dans la phase de préconception afin de fournir des estimations de coûts du système et des sous-systèmes sur l'ensemble du cycle de vie du produit.

Zhang et al. (Zhang and Gershenson 2003) détaillent ainsi trois métriques de coûts pour les coûts de fabrication, les coûts d'assemblage et les coûts de retrait.

Les coûts de fabrication correspondent à la somme des coûts des matériaux, de production (outillage et transformation) et des coûts d'achat.

Les coûts d'assemblage sont estimés en fonction du coût horaire de travail (\$/h) et des temps de manutention et d'assemblage manuels (h).

Les coûts de retrait concernent les coûts de recyclage, de réutilisation, de reconditionnement et de destruction.

- Le coût de recyclage est la somme des coûts de désassemblage, de broyage, de récupération des matériaux et des dépôts de déchets.
- Le coût de réutilisation s'évalue par l'équation suivante (1.4) :

$$\text{coût de réutilisation} = (1 + i)^T [rC_L(t_R + yt_D)]x \quad (1.4)$$

Avec i : taux d'actualisation

T : temps entre la fabrication et la réutilisation (h)

r : taux de composants retournés

C_L : coût horaire de travail (\$/h)

t_R : temps exigé pour les tests (h)

y : taux de récupération (%)

t_D : temps de désassemblage (h)

x : nombre initial de composants fabriqués

- Le coût de destruction est fonction du coût de dépôt d'ordures d'une tonne de déchets solides (\$) et du poids des déchets à jeter (tonne).

Les métriques de coûts sont naturellement à rapprocher des métriques de risques et de planning, dont elles sont fortement dépendantes par définition.

4.1.4 Les métriques de planning

La planification du projet établit la gestion et les infrastructures nécessaires pour évaluer et contrôler l'avancement d'un projet. Ce processus permet d'identifier avec un calendrier les besoins de ressources en personnel, en compétences et en installations, qu'elles soient internes ou externes à l'entreprise. Le processus de gestion du planning établit et détermine les procédures de gestion et l'exécution d'un effort technique, l'identification des tâches techniques, leurs interdépendances, les risques et les opportunités, et fournit des estimations en ressources nécessaires et budgétaires (DoD 2001). Cela détermine aussi les besoins en équipements spécialisés au cours du cycle de vie du projet pour améliorer l'efficacité du système et assurer une diminution des dépassements de coûts. A titre d'exemple lors de la conception du produit, divers spécialistes de disciplines travaillent ensemble pour évaluer les paramètres et les contraintes de planning associés à la fabricabilité, la testabilité, l'opérabilité et au maintien du produit (INCOSE 2006).

4.1.5 Les métriques d'efficacité (MoE)

Les Mesures d'Efficacité (ou MoE pour *Measures of Effectiveness*) sont des indicateurs techniques en terme d'efficacité opérationnelle. Un *Indicateur Technique* a pour rôle de préciser des critères issus d'exigences techniques allouées à un(e) ou plusieurs fonctions/composants. Ces critères ne sont généralement pas directement mesurables et sont construits à partir d'exigences de performances ou de paramètres de conception qui décrivent le fonctionnement global du système. L'efficacité opérationnelle est en général le degré de capacité d'un système à accomplir avec succès une mission en considérant l'environnement opérationnel global (DoD 2001), (Lo 2013). Autrement dit, elle évalue la relation entre les performances fournies par le système pendant la phase opérationnelle, et l'effort nécessaire correspondant de l'utilisateur, des ressources et de temps. Les MoE doivent prendre en compte la totalité des missions du système pour garantir que le système répondra à toutes les exigences fonctionnelles principales. Les MOE décrivent alors les capacités opérationnelles souhaitées d'un système en termes de résultats attendus et définissent ainsi la satisfaction d'une partie prenante.

Un MoE peut être composé d'une ou plusieurs mesures de performance (MoP) ou d'usage (MoS).

4.1.5.1 Les Métriques (ou Mesures) de Performance (MoP)

Les Mesures de Performance (ou MoP pour *Measures of Performance*) caractérisent les propriétés fonctionnelles et physiques du système relatives à l'exécution de la mission ou de la fonction du système (DoD 2001). Dans la phase de préconception, les métriques de performances permettent ainsi de mesurer l'adéquation du concept avec les performances attendues. Ceci suppose de savoir (Neely et al.):

- Spécifier qualitativement et quantitativement les exigences de performance des différentes fonctions de service du système dans les différents scénarii d'utilisation en fonction des hypothèses du cahier des charges;
- Allouer les exigences de performance aux sous-fonctions dans le cadre de la décomposition fonctionnelle;
- Dédire les caractéristiques de performance des organes de l'architecture organique et comparer différentes hypothèses d'architecture;
- Valider l'architecture physique en démontrant que les exigences de performance seront respectées.

Les MoP sont utilisées pour développer, supporter, et documenter les exigences de performance qui seront la base des activités de conception. Elles identifient aussi les paramètres techniques critiques qui seront tracés au travers des mesures de performance technique. En effet, un MoP est composé d'une ou plusieurs mesures de performance technique (TPM) (Lo 2013) (Chapitre 34.4.2.1).

A titre d'exemple dans le secteur automobile, un MoP peut être le nombre de secondes nécessaires pour un véhicule pour passer de 0 à 100 km/h, ou encore, le temps moyen entre deux pannes pour mesurer la performance du système en terme de fiabilité.

4.1.5.2 Les Métriques d'Usage/Conformité (MoS)

Les Mesures d'usage ou de conformité (ou anglais MoS pour *Measures of Suitability*) décrivent les capacités opérationnelles du système ou de disponibilité opérationnelle (« readiness »), en terme d'adéquation à l'usage, dans sa phase d'utilisation (DoD 2001).

Elles fournissent une information pertinente sur les conditions opérationnelles dans lesquelles le système doit fonctionner. L'évaluation du temps moyen entre deux défaillances (MTBF) en est un exemple : un système avec un taux de défaillances élevé peut par ailleurs satisfaire les autres exigences du système tout en étant impropre à l'usage. Ainsi, un exemple de mesures de conformité pourrait inclure une exigence de disponibilité opérationnelle supérieure à 90%.

4.2 Les métriques dans la phase de conception préliminaire

Dans la littérature, deux types de métriques pour la phase de conception préliminaire sont définis. Il s'agit des métriques portant sur la qualité des modèles et des données en général, et celles portant sur les choix de conception du système (par exemple les métriques pour comparer plusieurs architectures candidates).

Les premières, principalement issues de la communauté logicielle, permettent de juger de la qualité des différents modèles en les comparant. L'objectif étant alors de trouver d'une part quelles mesures détermineront si ce modèle est de meilleure qualité qu'un autre en fonction des règles/normes de qualité de l'entreprise, et d'autre part quels paramètres influenceront la qualité de la modélisation de conception (Heinrich et al. 2007; Blessing and Chakrabarti 2009). Dans son ouvrage, Kan (Kan 2002a) décrit les métriques et modèles d'ingénierie pour la qualité des logiciels : planification de la qualité, l'amélioration des processus et du contrôle de la qualité, de la gestion de qualité en cours, l'ingénierie de produits (conception et complexité des codes), l'estimation et prévision de la fiabilité, ainsi que l'analyse des données de la satisfaction du client. Il recense ainsi un grand nombre de types de métriques et de modèles pour l'ingénierie de la qualité des logiciels.

Les secondes permettant d'évaluer la conception d'un système, sont celles qui nous intéressent plus particulièrement dans le cadre de nos travaux, puisque ces métriques adressent des objectifs comme la modularité, la fiabilité, la complexité ... En effet, pendant la phase de conception préliminaire, les concepteurs cherchent les bonnes métriques d'évaluation pour retenir les configurations adéquates avant de se lancer dans des analyses plus détaillées souvent coûteuses (Hammadi 2012a). Par ailleurs, ces métriques sont cohérentes avec la démarche d'ingénierie système, en évaluant les solutions alternatives (choix d'architectures fonctionnelles, logiques ou physiques) et ainsi mieux gérer les compromis et les prises de décision.

4.2.1 Les métriques de modularité

« Un module peut être défini comme un élément physique d'un système ayant une interface clairement et explicitement définie, il est totalement autonome et fournit des fonctions bien identifiées » (Hehenberger 2012). Une autre définition est donnée par Ishii (Ishii 1998) : « un produit peut être vu comme un module s'il est composé d'un ensemble de sous-modules ».

Gershenson et al. (Gershenson et al. 2003) proposent une synthèse intéressante sur la modularité et ses avantages. L'objectif de modularité a été largement utilisé dans différents contextes allant de la conception à la fabrication de produits électroniques, mécaniques et de logiciels.

Hölttä-Otto et al. (Hölttä-Otto and de Weck 2007) présentent deux types de métriques pour la modularité, celle qui mesure l'indépendance des modules en mesurant le degré de couplage et celle qui identifie les similitudes entre les modules. Le premier type, pour mesurer le degré de couplages, s'appuie sur les relations entre les éléments d'une

architecture avec des couplages forts et faibles permettant de mesurer leurs degrés d'isolement. Il convient de souligner que les couplages entre les différents modules sont d'une grande importance spécifiquement pour les systèmes mécatroniques, puisque les interactions ne sont pas restreintes à une unique discipline, mais s'étendent à toutes les disciplines de la mécatronique. Par conséquent toutes les disciplines d'un système mécatronique doivent être traitées simultanément, idéalement dès le début du processus de conception (VDI-Fachbereich Produktentwicklung und Mechatronik 2004).

Dans le deuxième cas, l'objectif de la métrique de modularité est d'étudier la capacité d'un système à utiliser les mêmes matériaux, les mêmes processus de fabrication, les mêmes fournisseurs, la facilité de réutilisation ou de recyclage, etc. Cet objectif est supposé être atteint si les fonctions ou les composants avec les propriétés similaires sont regroupés ensemble dans un module. Ainsi, elle représente la capacité des modules à être réutilisés dans plus d'une architecture ou dans la conception d'autres produits.

Guo et al. (Guo and Gershenson 2004) proposent une comparaison de quatre métriques de modularité extraites de la littérature (Coulter et al. 1998), (Gu et al. 1997), (Stone et al. 2000), (Zhang and Gershenson 2003) calculées sur quatre produits industriels pour la reconception de ces produits.

S. Turki (Turki 2008a) a développé six indicateurs, implémentés avec le langage SysML, pour l'évaluation de la modularité et/ou de la réutilisation d'un composant mécatronique à partir d'architectures logiques de systèmes mécatroniques, modélisées par des diagrammes internes de bloc (ibd). Ainsi, un premier indicateur de centralisation permet par exemple, d'évaluer l'architecture de contrôle des composants, tandis que des indicateurs de couplage (architectural et fonctionnel) mesurent la dépendance des sous-systèmes entre eux. Un indicateur d'autonomie d'un sous-système est défini en fonction du nombre de liens strictement internes, du nombre total de liens et du nombre de composants constituant le système. Finalement, un indicateur d'interactivité est formalisé en fonction des entrées et des sorties du composant, il permet d'estimer les impacts des changements apportés à un modèle d'architecture et de pouvoir ainsi comparer plusieurs architectures.

Pourtant, bien que les métriques de modularité aient été largement développées pour la conception de produits industriels au cours de ces deux dernières décennies, il existe ni méthode, ni méthodologie systématique éprouvée qui aiderait les concepteurs à accroître la modularité d'un produit (Guo and Gershenson 2003).

4.2.2 Les métriques de fiabilité

La fiabilité d'un système est définie dans la norme NF X 060-010 (NF 1984), par la probabilité que le système accomplisse sans défaillance une fonction requise pendant une durée donnée et dans des conditions d'environnement et d'utilisation donnée.

Les mesures de fiabilité sont donc principalement basées sur des fonctions de probabilité. Soient λ le taux de défaillance et T la variable qui, à tout produit associe sa durée de vie ou son temps de bon fonctionnement, c'est à dire la durée jusqu'à l'instant où apparaît la première défaillance. Soit l'instant ($t = 0$) choisi comme instant de mise en service de l'appareil, on définit les fonctions suivantes :

- La fonction de probabilité de défaillance : $F(t)$ est définie pour tout $t \geq 0$ par

$$F(t) = P(T \leq t) = 1 - e^{-\lambda t} \quad (1.5)$$

Le nombre $F(t)$ représente la probabilité qu'un dispositif choisi au hasard dans la population ait une défaillance avant l'instant t .

- La fonction fiabilité $R(t)$ est définie pour tout $t \geq 0$ par

$$R(t) = 1 - F(t) = e^{-\lambda t} \quad (1.6)$$

Ainsi, le nombre $R(t)$ représente la probabilité qu'un dispositif choisi au hasard dans la population, n'ait pas de défaillance avant l'instant t .

- La densité de probabilité $f(t)$ d'avoir une défaillance (variable T définie précédemment) est définie par :

$$f(t) = \lambda e^{-\lambda t} \quad (1.7)$$

- La fonction taux de défaillance $\lambda(t)$ à l'instant t est définie par :

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{f(t)}{1 - F(t)} = \lambda \quad (1.8)$$

- MTTF (Mean Time To Failure) est le temps moyen avant la première défaillance et MTBF (Mean Time Between Failures) est le temps moyen entre deux défaillances consécutives :

$$MTBF = E(T) = \int_0^{+\infty} t \cdot f(t) \cdot dt = \frac{1}{\lambda} \quad (1.9)$$

La défaillance peut être évaluée par l'attribution d'une note sur les 3 critères suivants (Meinadier 1998) :

- La fréquence F marque la probabilité que la cause de défaillance se produise.
- La gravité est l'impact des effets des défaillances sur le système, ou une estimation de la sévérité de l'effet maximum de la défaillance.
- Le risque de non-détection marque la probabilité que la cause et le mode de défaillance étant apparus, la défaillance atteigne l'utilisateur.

Une métrique de fiabilité est donc fonction du taux de défaillance qui résulte du produit des critères décrits précédemment (Zwingmann 2005):

$$\text{Taux de Défaillance} = \text{Fréquence} \times \text{Gravité} \times \text{Détection} \quad (1.10)$$

Les paramètres influents pour évaluer la fiabilité (et donc la défaillance) peuvent avoir différentes origines. La Figure 1-29, ci-dessous illustre les causes non exhaustives de défaillance affectant la fiabilité dans la conception :

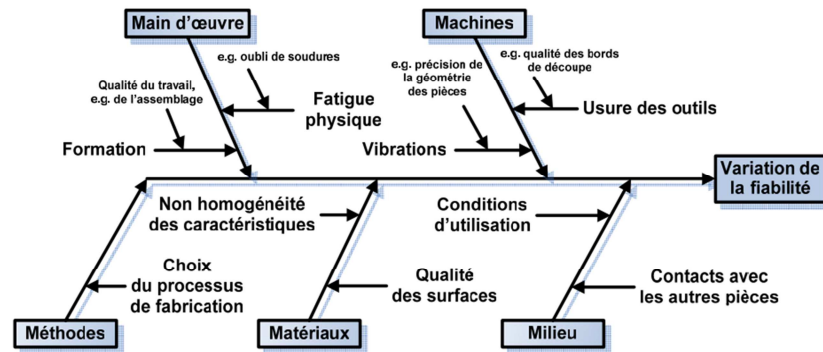


Figure 1-29 : Identification non exhaustive des paramètres influents dans l'évaluation de la fiabilité (Zwingmann 2005).

4.2.3 Les métriques de complexité

Tandis que les concepts et mesures de complexité ont largement été publiés dans la littérature, quel que soit le domaine. P. Hehenberger (Hehenberger 2012) propose pour la mesure de complexité des systèmes mécatroniques, de considérer les aspects suivants :

- la description et l'évaluation de tous les couplages entre les différentes disciplines et les paramètres de conception ;
- la structure (hiérarchique) d'un système mécatronique ;
- les fonctions et leur matérialisation.

Ainsi elle peut être mesurée suivant les aspects de connectivité et de variété des éléments. Il détaille ainsi la complexité de conception mécatronique suivant 5 catégories : fonctionnelle, structurale, mécatronique (multidisciplinaire), d'interfaces (E/S) et de dépendances (des paramètres).

Nous pouvons également citer quelques métriques de complexité proposées par d'autres auteurs.

La complexité de conception

La complexité d'un système pendant la préconception peut être définie en s'inspirant de l'approche axiomatique proposée par Suh (Suh 1990) détaillée au Chapitre 12.3.4.4, comme l'incertitude dans la réalisation des exigences fonctionnelles. En effet, selon le 2^{ème} axiome de cette approche, la meilleure conception est définie comme celle qui contient le minimum d'information et correspond donc à une complexité minimale.

Elle s'exprime par la relation suivante :

$$I_i = \log_2 \left(\frac{1}{P_i} \right) = -\log_2(P_i) \quad (1.11)$$

Avec I_i le contenu d'information

P_i la probabilité de succès des exigences fonctionnelles

Cette métrique évalue la complexité par rapport à l'adéquation de l'architecture physique par rapport aux exigences fonctionnelles.

La complexité (fonctionnelle) du produit

A partir de la décomposition fonctionnelle d'un problème, Bashir (Bashir and Thomson 1999) définit la complexité d'un produit (PC : Product Complexity) par le nombre de fonctions et la profondeur de leur arbre de décomposition fonctionnelle (hiérarchies). La métrique s'exprime par la relation suivante :

$$PC = \sum_{j=1}^l j \cdot f_j \quad (1.12)$$

Avec

$$f_j = \sum_{k=1}^{F_j} w_k \quad (1.13)$$

Où w_k est le poids assigné à la fonction k du niveau j , F_j est le nombre de fonctions à un niveau j et l le nombre de niveaux.

La complexité mécatronique

Smailagic (Smailagic et al. 1995) définit la complexité mécatronique par le produit de la complexité de différentes disciplines impliquées. La complexité de chaque discipline étant mesurée par des critères spécifiques à cette discipline :

$$\begin{aligned} \text{Complexité mécatronique} \\ = \text{complex.mécanique} \times \text{complex.logicielle} \\ \times \text{complex.électronique} \end{aligned} \quad (1.14)$$

Cependant, la méthode d'évaluation de la complexité pour chaque discipline n'est pas détaillée. Ce n'est pas une métrique au sens où elle ne respecte pas les règles que nous avons définies et par conséquent, ne permet pas de comparer la complexité entre plusieurs architectures candidates.

La complexité (structurale) de conception

Un système mécatronique peut être décomposé en structure hiérarchique (*Annexe 4*) (Hehenberger 2012). Toute structure hiérarchique comprend des blocs qui sont connectés par des branches. Le nombre de niveaux de décomposition et le nombre d'éléments utilisés sont des indicateurs pour la complexité de la conception. Une métrique peut alors être définie comme :

$$\text{Complexité de conception} = \sum_{j=1}^l j \times N_j \quad (1.15)$$

Avec

N_j : le nombre d'éléments mécatroniques au niveau j .

l : le nombre de niveaux dans l'arbre.

Cette métrique permet d'évaluer la complexité des concepts mécatroniques dès les premières phases de conception, mais ne prend pas en compte les interactions multi-domaines.

La complexité (taille, couplages et résolabilité)

Summers et al. (Ameri et al. 2008; Summers and Shah 2010) propose des métriques de complexité adressant les 3 points de vue suivant :

- la taille (correspondant à la quantité d'informations liées à la conception)

$$\begin{aligned} C_{x_{size-process}} = & \left((M^0 + C^0 + P_{op}) \right. \\ & \times \ln |idv + ddv + dr + mg + a_{op} + e_{op} + s_{op} \\ & \left. + r_{op}| \right) \end{aligned} \quad (1.16)$$

- les couplages ou interactions entre les éléments, basés sur un algorithme qui traitent des graphes de connexion :

$$level \times set\ size \times number\ of\ sets + total \quad (1.17)$$

- la résolubilité d'un problème de conception, prenant en compte l'effort de conception et les degrés de liberté du problème.

$$C_{x_{solvability}} = \sum (k_1 \cdot a_{op} + k_2 \cdot s_{op} + k_3 \cdot e_{op} + k_4 \cdot r_{op}) \quad (1.18)$$

$$C_{x_{DOF_{prob}}} = \left(\sum DOF(idv) + \sum DOF(ddv) + \sum DOF(mg) - \sum DOF(dr) \right) \quad (1.19)$$

4.3 Les métriques dans la phase de conception détaillée

Les métriques en phase de conception détaillée évaluent des objectifs en relation avec la géométrie, le positionnement des composants et leurs caractéristiques physiques détaillées. Elles sont principalement basées sur l'architecture physique et l'architecture spatiale et peuvent prendre en compte les flux et couplages multi-physiques. Ces métriques, que ce soit pour la maintenabilité, l'assemblage, le démontage, la recyclabilité, etc, ont en commun qu'elles font intervenir les propriétés géométriques ou la position des composants. Ainsi, elles sont très souvent liées les unes aux autres et les compromis sont donc souvent difficiles.

Considérons les métriques de maintenabilité. La maintenabilité évalue la facilité d'un système à maintenir le système pendant sa phase d'utilisation. Le processus de maintenance comprend donc des activités fournissant un soutien opérationnel, logistique et de gestion du matériel. Une fois que les problèmes sont identifiés, des mesures correctives ou préventives sont prises pour rétablir la capacité du système complet (INCOSE 2006). D'après la norme française NF X 60-010 (NF EN 13306 2010), la maintenance regroupe l'ensemble des activités destinées à maintenir ou à rétablir un système dans un état ou dans des conditions données de sûreté de fonctionnement pour accomplir une fonction requise. De plus, elle définit la maintenabilité comme une probabilité qu'un système, lorsqu'il nécessite une intervention de maintenance soit remis dans un état de fonctionnement donné, dans des limites de temps spécifiées et lorsque le travail est effectué selon des procédures prescrites et des conditions données. Elle doit s'assurer des possibilités d'intervention sur le système.

D'après (Wani and Gandhi 1999), la maintenabilité peut être estimée à partir de douze indicateurs regroupés dans le tableau ci-dessous (Table 1-2)

1	Accessability	7	Modularization
2	Disassembly/assembly	8	Tribo-concepts
3	Standardization	9	Personnel including ergonomics
4	Simplicity	10	System environment
5	Identification	11	Tools and test equipment
6	Diagnosability	12	Documentation

Table 1-2 : Liste de critères pour évaluer la maintenabilité (Wani and Gandhi 1999)

Chaque indicateur est évalué manuellement par une valeur comprise entre 0 et 4, et la métrique de maintenabilité est alors calculée par le permanent de la matrice (utilisé pour la répartition des éléments) de couplage de ces indicateurs, créée à l'aide d'un graphe orienté. Ce graphe détermine le niveau de « facilitation » de chaque indicateur par rapport à un autre : par exemple l'accessibilité facilite le désassemblage. L'accessibilité permet ainsi de réduire le temps d'intervention de maintenance.

D'autres auteurs ont étudié la maintenabilité et notamment certains indicateurs décrits dans la Table 1-2, dont nous citerons quelques exemples ci-dessous.

Das et al. (2000) (Das et al. 2000a) proposent un indicateur pour mesurer le désassemblage qui repose sur 7 critères : le temps, l'outillage, la technique d'enlèvement, l'accessibilité, les instructions de désassemblage, les protections nécessaires et l'effort. Chaque critère a une échelle quantitative qui lui est propre (Figure 1-30) et est évalué manuellement. Cet indicateur est compris en 0 et 100. Le critère de l'accessibilité comprend par exemple, la visibilité, les mouvements composés, l'accès par-dessus, la profondeur > 15 cm, le dégagement horizontal, l'accès par-dessous. Ces différents aspects peuvent être pondérés (Zwingmann 2005).

1. TIME (Secs)	>210	140	90	50	25	<5	SCORE
	25	20	15	10	5	0	
2. TOOLS	Improvised	Special	OEM	Mechanic	Air Gun	None	SCORE
	10	8	6	4	2	0	
3. FIXTURE	Automation	Winch	Clamps	Two-Hands	One-Hand	None	SCORE
	15	12	9	6	3	0	
4. ACCESS	Not Visible	Dual-axis	From below	> 6" deep	X/Y-axis	Z-axis	SCORE
	15	12	9	6	3	0	
5. INSTRUCT	Training	Contact OEM	Group Discuss	> 30 secs	10-20 secs	None	SCORE
	10	8	6	4	2	0	
6. HAZARD	Body-suit	Air Supply	Fire Protection	Face Mask/Arm wrap	Gloves	None	SCORE
	5	4	3	2	1	0	
7. FORCE - Unfasten - Human - Machine	High-Impact	Low-Impact	Leverage	Orthogonal	Torsional	Axial	SCORE
	> 50lbs	35	24	15	7	2	
	> 300lbs	220	160	110	75	50	
	20	16	12	8	4	0	

Figure 1-30 : Exemple d'échelle pour quelques critères (Das et al. 2000a)

En effet l'accessibilité se définit par la caractéristique d'un élément à être atteint facilement, comme par exemple un élément se situant à l'intérieur d'un autre et qui serait aisément atteint de l'extérieur. L'accessibilité permet de réduire le temps d'intervention et donc de réparation, améliorant ainsi la disponibilité et par conséquent la maintenabilité du produit. Selon la norme NF X 60-301 (NF 1982), un critère permettant d'évaluer l'accessibilité

correspond à la rapidité avec laquelle un élément peut être atteint. Cette rapidité est fonction de l'importance qualitative et quantitative des travaux connexes à consentir pour avoir le libre accès et remettre l'ensemble dans son état initial.

Hung-Yao Hsu (Hsu and Lin 2002) considère l'accessibilité comme une mesure quantitative (CA) de l'influence de la présence d'autres composants sur un composant, lorsque celui-ci se rapproche de sa destination avec une orientation fixe le long d'un parcours en ligne droite sans collision à l'intérieur des limites du produit. Il considère le composant en phase de préconception comme une boîte aux formes simplifiées (Figure 1-31).

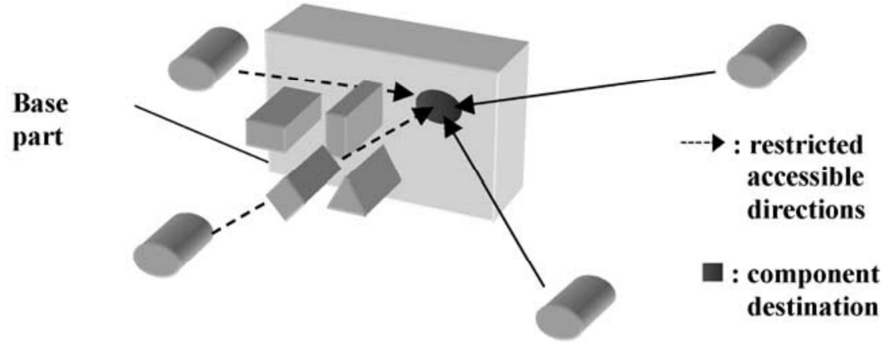


Figure 1-31 : Restriction d'accessibilité sur certaines directions en présence d'autres composants (Hsu and Lin 2002)

Cette mesure prend en compte le volume interdit (FV_k : « Forbidden Volume ») correspondant à la projection de l'obstacle sur la demi-sphère du volume accessible idéal (Figure 1-32 i). Pour calculer ce volume interdit, Hung-Yao Hsu décompose son polyèdre sphérique (Figure 1-32 ii) en deux pyramides sphériques OABC et OACD dont il utilise les longueurs des côtés a_k , b_k et c_k .

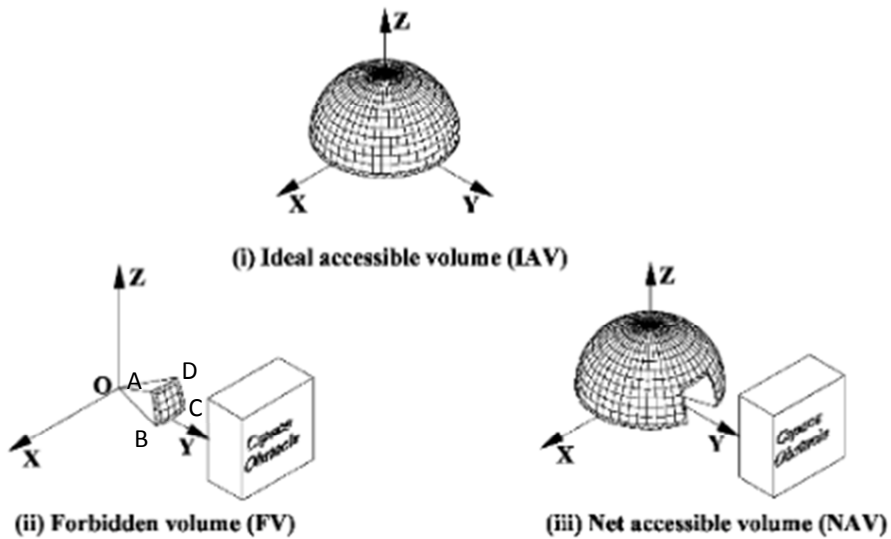


Figure 1-32 : Volume accessible d'un composant et détail de la décomposition de ce volume (Hsu and Lin 2002)

$$CA = \frac{1}{\log_2 \left(\frac{\frac{2\pi}{3}}{\frac{2\pi}{3} - U_{j-1}^n (U_{k-1}^2 FV_{jk})} \right)} \quad (1.20)$$

U : union de l'ensemble des opérations

n : nombre d'obstacles

j : l'espace de configuration j

k : le k ième triangle sphérique

Avec FV_k : (« Forbidden Volume ») volume interdit associé à chaque obstacle k .

$$FV_k = \frac{4}{3} \tan^{-1} \left(\tanh \left(\frac{S_k}{2} \right) \tan \left(\frac{S_k - a_k}{2} \right) \tan \left(\frac{S_k - b_k}{2} \right) \tan \left(\frac{S_k - c_k}{2} \right) \right) \quad (1.21)$$

$k = 1, 2$

$$S_k = \frac{a_k + b_k + c_k}{2}$$

a_k, b_k, c_k sont les longueurs des côtés du triangle sphérique ABC.

Fujimoto et al. (Fujimoto et al. 2001) décrivent également l'accessibilité pour des phases de maintenance, comme la facilité d'accès à un composant. Ils définissent quatre indicateurs : le degré de liberté de la pièce (E_{df}), la possibilité angulaire de retrait (E_{dir}), l'espace pendant le retrait de la pièce (E_{sup}) et l'angle de retrait (E_{ds}).

- La mesure du degré de liberté de la pièce (E_{df}) se réfère à l'espace dans lequel le composant peut être déplacé librement tout en le séparant du reste du corps (Figure 1-33), avec comme hypothèse que le centre de l'objet coïncide avec le centre du système de coordonnées.

$$E_{df} = - \sum_i^{x,y,z} \log_2 \frac{\Delta_i}{180} \quad (1.22)$$

L'angle Δ_i représente le degré de liberté du composant dans la configuration actuelle tandis que le dénominateur 180 représente le degré de liberté de l'objet lorsqu'il n'y a pas d'autres corps autour en collision avec lui. Cela signifie que le composant perd en degré de liberté par la présence d'un autre corps autour de lui.

- La mesure de la possibilité angulaire (E_{dir}) de retrait représente la direction dans laquelle le composant doit être retiré. θ mesure l'angle de retrait du composant par rapport à la verticale ascendante (Figure 1-33).

$$E_{dir} = - \frac{180 - \theta}{180} \log_2 \frac{\Delta}{180} \quad (1.23)$$

Cette mesure est directement liée à celle de la précédente E_{df} , puisque la possibilité de retrait est liée de fait au degré de liberté de la pièce.

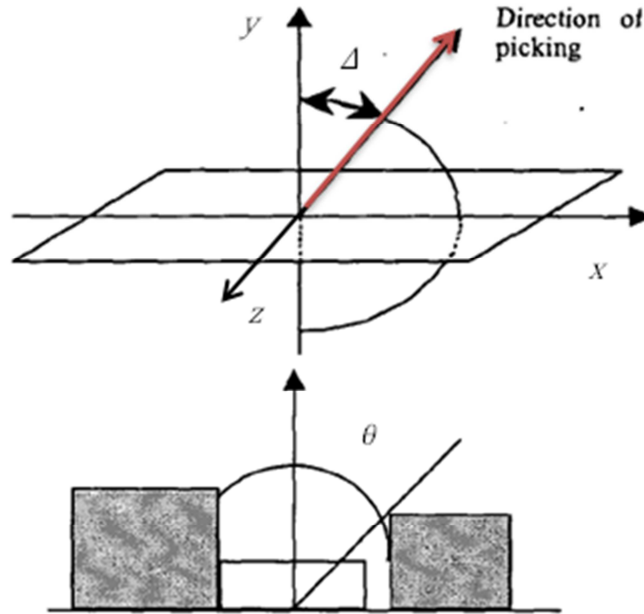
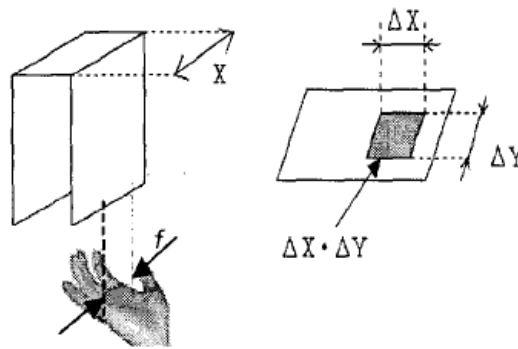


Figure 1-33 : Degré de liberté et direction du composant (Fujimoto et al. 2001)

- La mesure de l'espace nécessaire (E_{sup}) pour retirer un composant ou réaliser n'importe quelle opération en relation avec le retrait d'un composant, s'exprime par la relation suivante :

$$E_{sup} = - \left(\log_2 \frac{\Delta X}{X} + \log_2 \frac{\Delta Y}{Y} + \log_2 \frac{\Delta Z}{Z} \right) \quad (1.24)$$



.One dimensional case 2 dimensional case

Figure 1-34 : Espace nécessaire pour le retrait des pièces (Fujimoto et al. 2001)

- La mesure de l'angle de retrait (E_{ds}) indique l'importance d'examiner la direction dans laquelle le composant doit être pris en charge. En effet, même s'il y a assez de place pour ôter un composant, il est difficile d'effectuer l'opération si la direction n'est pas appropriée.

$$E_{ds} = - \frac{180 - \phi}{180} \log_2 \left(\frac{\Delta X \cdot \Delta Y \cdot \Delta Z}{X \cdot Y \cdot Z} \right) \quad (1.25)$$

Guide et al. (Guide et al. 1997) proposent un indicateur pour mesurer la complexité de réassemblage (C) en fonction du nombre de pièces à coordonner. Ils définissent (C) comme étant le produit de la complexité de réassemblage (C_a) par la complexité d'arborescence de l'assemblage (C_d) et la complexité des opérations (C_r).

$$C = C_a \times C_d \times C_r \quad (1.26)$$

La valeur de cet indicateur est comprise entre 0 et 1. Plus la valeur de cet indicateur est grande, plus le produit est complexe à assembler.

La complexité de réassemblage (C_a) est définie par l'expression suivante :

$$C_a = \frac{NU}{NU_{max}} \quad (1.27)$$

Avec

NU : Nombre d'éléments qui doivent être coordonnés pour effectuer l'assemblage d'un produit

NU_{max} : Nombre maximum d'éléments à coordonner pour réassembler n'importe quel produit pris en compte par l'entreprise.

La complexité de profondeur d'arborescence (C_d) est définie par l'expression suivante :

$$C_d = \frac{NL}{NL_{max}} \quad (1.28)$$

Avec C_d : le taux de complexité de profondeur pour un produit défini.

NL : le nombre de niveaux dans la nomenclature de ce produit.

NL_{max} : le nombre maximum de niveaux dans la nomenclature pour tous les produits.

La complexité des opérations (C_r) est définie par l'expression suivante :

$$C_r = \frac{Max\{R\}}{NR_{max}} \quad (1.29)$$

Avec

$Max\{R\}$: Nombre maximum d'opérations de routage exigées pour une pièce d'un produit

NR_{max} : Nombre maximum d'opérations pour tous les produits pris en compte par l'entreprise.

Lors d'une tâche de montage/désassemblage ou de maintenance, les trajectoires où les pièces considérées peuvent être montées ou extraites sans collision, doivent être générées. Les travaux de Zwingmann (Zwingmann 2005) recensent les méthodes de quantification de l'effort de désassemblage (Figure 1-35).

Effort déterminé par	Exemples de travaux ⁶
Expérience/Historique	
- Pondération de difficultés opératoires (type de liaisons, etc.)	[2], [4], [26], [30], [31], [35], [37], [42], [43], [44], [57]
- Pondération de la dextérité requise	[23]
- Coûts, énergies ou durées d'opérations connus	[1], [2], [6], [15], [20], [21], [22], [27], [39], [46], [48], [49], [51], [55], [58]
- Fonction de pénalisation des durées	[5], [22], [27], [30], [55]
- Décomposition des mouvements	[38], [53]
Mesure de l'entropie	
- Accessibilité (direction d'extraction, ...etc.)	[25], [52], [56]
- Accessibilité des outils (orientation, ...etc.)	[24]
- Nb. d'op. de séparations pour isoler la cible	[51]
Caractéristiques du produit (via la CAO)	
- Nb. pièces	[7], [50]
- Accessibilité par le volume	[7], [50]
- Accessibilité par les directions d'extraction ⁷	[8], [9], [11], [12], [16], [17], [28], [29], [41]
- Matériaux dangereux	[7], [22], [27]
- Force requise pour défaire les liaisons	[36], [56]
- Stabilité des sous-assemblages	[3], [33], [34], [35], [36], [37], [45]
- Mobilité des sous-assemblages	[35]
Caractéristiques des opérations	
- Nb. de réorientations	[3], [13], [14], [22], [27], [34], [47]
- Nb. changt. d'outils	[22], [27], [40]
- Nb pièces à retirer	[10], [11], [12], [14], [19], [40], [41], [54]
- Ergonomie, Force	[18], [30], [31], [32], [40]
- Visibilité	[40]
- Durée obtenue par simulation	[32]

Figure 1-35 : Synthèse des différentes méthodes pour évaluer l'effort de désassemblage (Zwingmann 2005)

Zwingmann propose un autre critère pour la démontabilité. Il évalue pour toutes les pièces du composant c , la nature des attaches et la complexité d'intervention par l'expression suivante :

$$\begin{aligned}
 & demonte (Composant\ c, Pièce\ p1) \\
 &= \sum_{Pièce\ p2 \in composant\ c, p2 \neq p1} [Nature\ attache\ (Pièce\ p1, Pièce\ p2) \\
 &\quad \times Complexité(Composant\ c, Pièce\ p1)] \\
 &\quad \times ChangementOutil\ (Composant\ c, Pièce\ p1) \\
 &\quad \times PénalitéChangementOutil \\
 &\quad \times ChangementDirections\ (Composant\ c, Pièce\ p1) \\
 &\quad \times PénalitéChangementDirection
 \end{aligned} \tag{1.30}$$

A partir d'informations issues d'un fichier de CAO, Mani et al. (Mani et al. 2001) créent aussi une métrique permettant d'évaluer le coût de désassemblage pour un sous-ensemble Ψ_s . Cette métrique est composée de trois indicateurs.

Le premier est l'indicateur de « facilité » de démontage L_s . Cet indicateur permet d'estimer le nombre de liens de fixation par sous-ensemble par la relation suivante :

$$L_s = 1,25(A_s - 1) + 1 \tag{1.31}$$

Où

$$A_s = \sum_i (X_i | P_i = s) \tag{1.32}$$

Avec A_s : le nombre de pièces démontées dans le sous-ensemble

X_i : variable 0 ou 1 indiquant le démontage d'une pièce

P_i : sous-ensemble d'une pièce

s : sous-ensemble dans le produit

Le deuxième indicateur est un indicateur de compacité δ_s . Plus la densité du sous-ensemble augmente et plus l'accessibilité diminue. Par conséquent, le démontage des pièces devient plus difficile. Cet indicateur est fonction du nombre de pièces démontées, du volume

« enveloppe » du produit et de la somme des volumes « enveloppe » de chaque pièce. Il s'évalue par la relation suivante :

$$\delta_s = e^{\frac{(2-G_s)}{5}} \quad (1.33)$$

Avec

$$G_s = \frac{V_{prod}}{A_s \times V_s} \quad (1.34)$$

V_{prod} est l'enveloppe du volume produit

V_s est la somme des enveloppes pour chaque partie dans le sous-ensemble.

Le dernier indicateur est celui de l'encapsulation (l'imbrication) γ_s . L'effort de démontage augmente proportionnellement à l'augmentation des niveaux d'imbrication. Cet indicateur est fonction du nombre de pièces démontées, du niveau dans lequel se situe le sous-ensemble et le nombre de pièces du sous-ensemble.

La métrique s'écrit alors sous la formule suivante :

$$\Psi_s = \alpha \{L_s(1 + \delta_s) + \gamma_s\} \quad (1.35)$$

α étant un facteur de coût moyen de désassemblage.

Malheureusement, ces indicateurs ont des facteurs de pondération non expliqués, et le raisonnement pour aboutir à ces indicateurs n'est pas exposé.

La recyclabilité est liée à la fin de vie d'un système, toutefois il est important de prévoir cet aspect dès sa conception. Les métriques de recyclabilité font appel à des critères comme l'accessibilité, la démontrabilité (décrits précédemment) mais également le coût et l'efficacité des opérations de recyclage.

A ce propos, nous pouvons citer les travaux de Atlee (Atlee and Kirchain 2006) proposant une métrique pour analyser l'efficacité de récupération d'une opération de recyclage à l'aide de l'expression suivante :

$$\text{Indicateur de recyclabilité de la valeur} = \frac{\sum_j V_{p_j} \times m_j}{\sum_i \sum_k V_{m_{ki}} m_{ki}} \quad (1.36)$$

Avec (Figure 1-36) V_p valeur du matériau pour le produit final après recyclage
 V_m valeur du matériau avant la transformation en production
 m masse des matériaux
 i indice du matériau entrant
 j indice du matériau sortant
 k $k^{\text{ième}}$ matériau

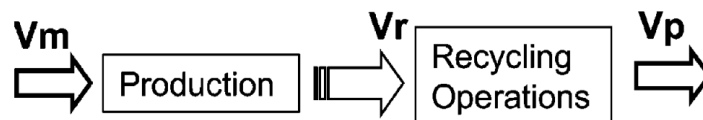


Figure 1-36 : Paramètres de recyclabilité (Atlee and Kirchain 2006)

Cette mesure indique la capacité à récupérer non seulement la masse mais aussi la valeur des matériaux du produit en fin de vie (V_p) par rapport à la valeur des matériaux qui ont été

initialement consommés (V_m). La mesure de la valeur de rétention donne un aperçu de l'efficacité de recyclage du système.

De façon similaire, l'auteur donne un indicateur de la recyclabilité de l'énergie, à l'aide de la relation suivante :

$$\begin{aligned} & \text{Indicateur de recyclabilité de l'énergie} \\ &= \frac{\sum_j \sum_k (E_{p_{kj}} - E_{r_{kj}}) \times m_{kj}}{\sum_i \sum_k E_{max_{ki}} \times m_{ki}} \end{aligned} \quad (1.37)$$

Avec E_p l'énergie totale utilisée pour la matière première

E_r l'énergie totale utilisée pour la matière transformée

E_{max} l'énergie totale utilisée maximale

Il est à noter que dans un rapport de la société WRAP (WRAP 2008), d'autres métriques de recyclabilité sont définies selon d'autres point de vue, comme celui de l'empreinte carbone. Si un produit A a une empreinte carbone représentée par X (kg/kg) et qu'il peut être recyclé en un produit B qui consomme du dioxyde de carbone Y (Kg/Kg) alors que le produit B vierge aurait consommé Z (Kg/Kg) de dioxyde de carbone, le taux de recyclabilité peut s'évaluer par la relation suivante :

$$\text{recyclabilité de dioxyde de carbone} = \frac{Z - Y}{X} (\%) \quad (1.38)$$

Une autre métrique est celle de la recyclabilité par le poids. Si X tonnes d'un produit A peuvent être recyclées en un produit B et pendant ce processus Y tonnes de matériaux sont perdues, alors la métrique de recyclabilité par le poids s'écrit :

$$\text{recyclabilité par le poids} = \frac{X - Y}{X} (\%) \quad (1.39)$$

4.4 Les métriques en phase de vérification et validation

Par définition, les phases de vérification et de validation sont des phases d'évaluation. En effet, il est important de s'assurer que la conception est correcte aux vues des exigences de conception (vérification) et que le produit conçu correspond bien aux exigences des parties prenantes (validation).

Généralement, ces métriques/critères sont définis au début du projet dans des documents de spécifications qui détaillent comment seront évalués la conception et le produit, de sorte à être validés avant le début de la fabrication. Elles sont basées sur l'ensemble du cycle de conception, sur les variables de conception et les liens ou dépendances avec les exigences.

Ainsi, ces métriques permettent de mesurer l'adéquation ou l'écart avec les exigences et aident à la traçabilité du dossier de conception.

4.4.1 Les métriques de vérification

4.4.1.1 Les métriques de contrôle du processus

Les métriques de gestion/contrôle des processus sont des mesures prises pour suivre le processus de développement, de réalisation et le déploiement du système. Elles comprennent un large éventail de paramètres influents qui peuvent être la disponibilité des ressources, le taux du temps d'activité, les tâches terminées, le taux d'achèvement et les

satisfactions de l'équipe (DoD 2001). D'autre part, pour l'amélioration des processus, un indicateur d'efficacité du processus de développement est par exemple le ratio entre la proportion de reconception et la charge totale de conception ; à titre d'exemple en comparant pour un projet donné sur une période donnée, les heures passées pour corriger les défauts et la somme des heures de travail. En mesurant l'évolution avant et après, les équipes mesurent l'influence de l'amélioration du processus (Tonnellier and Terrien 2012). Le coût unitaire de production estimé peut également être suivi pendant l'effort de production (*métriques de coût*).

Les *métriques de planning* évaluent la progression et l'atteinte de jalons du projet. Elles mesurent l'écart entre le planning prévu et le déroulement actuel du projet.

Les *métriques d'usage* (« suitability ») relatives à l'aptitude opérationnelle, précédemment décrites pour la phase de préconception, peuvent être également utilisées comme des métriques de vérification, en indiquant la progression du processus de conception vers un système opérationnel ad hoc.

Les métriques de processus peuvent aussi inclure des mesures qui indiquent l'amélioration dans la productivité, la testabilité, le degré de simplicité de conception ou la robustesse de la conception. Nous pouvons citer pour illustrer nos propos le cas des mesures du nombre de pièces suivies, du nombre de pièces similaires, et du nombre de pièces d'usures, elles fournissent des indications de productivité, de maintenabilité et de simplicité de conception.

4.4.2 Les métriques de validation

4.4.2.1 Les Mesures de Performance Technique (TPM)

Les Mesures de Performance Technique (ou TPM pour Technical Performance Measurement) sont des indicateurs techniques dont la valeur est donnée par la valeur d'un paramètre de conception du système. Les valeurs des TPM permettent d'apprécier, tout au long du cycle de conception, le niveau de satisfaction d'une exigence technique par une mesure de performance (MoP). Les TPM évaluent le progrès de conception, concourent à la surveillance et à la traçabilité des risques techniques. Un TPM est généralement discuté avec les parties prenantes au début du projet et est quantifié ou qualifié par un paramètre influent de conception (Roedler et al. 2005).

Ces exigences techniques peuvent par exemple concerner la fiabilité, la puissance, le poids, le débit, la complexité, la vitesse, ou la précision.

TPM	Planned value	Current value	Relative importance
Image processing time (sec.)	30 sec. max	45 sec. for an SES simulation	10
Power required	10 KV max. UPS 2 hr. backup	12 KV UPS 1.5 hr. Vendor data	8
Weight	600 lbs. max man portable modules	625 lbs. six-modules CAD mockup	7

Figure 1-37 : Exemple d'indicateurs TPM (Oakes et al. 2006)

4.4.2.2 Les métriques d'usage « Suitability metrics »

La métrique d'usage est la mesure pour laquelle un système peut être mis en service de manière satisfaisante, considérant certains objectifs comme la disponibilité, la fiabilité, la maintenabilité, la sécurité, la documentation, la main d'œuvre, etc. (DoD 2001), (Hussain and Ferneley 2008), (Seffah et al. 2006) et (ISO 2002). Si ces objectifs sont des exigences des parties prenantes, alors ces métriques deviennent des métriques de validation.

4.4.2.3 Les métriques de satisfaction

Les métriques de satisfaction sont un objectif clé pour l'entreprise, puisqu'elles adressent la qualité perçue.

Clivillé (Clivillé 2004) propose un exemple de métriques de satisfaction client. La métrique est décomposée en 5 indicateurs. Il s'agit de :

- L'indicateur de conformité des livraisons exprimé par la relation suivante :
$$\frac{\text{nombre de commande livrées conformes et dans les délais}}{\text{nombre de commandes}} \quad (1.40)$$

- L'indicateur de qualité du produit évalué par l'expression suivante :
$$\frac{\text{nombre de produits échangés}}{\text{nombre de commande}} \quad (1.41)$$

- L'indicateur de l'offre du produit qui est défini en fonction de la largeur et de la profondeur de la gamme

- L'indicateur du délai de livraison qui s'écrit alors :
$$\frac{\text{nombre de semaines de livraison}}{\text{nombre moyen de semaines de livraison}} \quad (1.42)$$

- L'indicateur du climat social qui est exprimé en fonction des informations diverses sur le climat social.

4.4.2.4 Les métriques de traçabilité

Les métriques de traçabilité sont généralement issues des matrices de traçabilité, et évaluent et tracent le degré de satisfaction des exigences clients aux exigences de niveau inférieur avec un ensemble de liens documentés. La traçabilité fournit des informations qui aident à déterminer si toutes les relations et dépendances sont adressées.

Limon et al. (Espinoza Limón and Garbajosa Sopena 2005) se sont penchés sur les caractéristiques de l'analyse et de l'évaluation de la traçabilité :

- Les liens peuvent porter soit sur le processus soit sur le produit.
- Des catégories de relations de traçabilité pré- et post- spécification des exigences sont définies.
- L'objectif du lien de traçabilité doit fournir les informations sur la raison ou la justification de l'existence du lien afin de justifier les ressources du projet qui seront dépensées (les ressources telles que, par exemple, la surveillance, la sauvegarde et l'analyse du projet afin de prendre des décisions, etc.)
- Les articles ou les objets que le lien de traçabilité devra relier. Il est souhaitable que les objets puissent être définis spécifiquement, afin de rendre le plus automatique possible, les tâches de liaison.

Finalement, Ali définit cinq types de mesures de traçabilité (Ali 2006) :

- La première concerne les exigences spécifiées à un niveau inférieur et/ou supérieur.
- La deuxième est similaire à la première mais elle mesure les liens de traçabilité sur l'ensemble des niveaux et pas seulement sur les niveaux immédiatement supérieur ou inférieur.

- La troisième métrique inventorie le nombre d'exigences qui ont des liens incompatibles dans les deux directions.
- La quatrième correspond au nombre d'exigences qui n'ont pas de liens de traçabilité au niveau inférieur ou supérieur.
- La dernière mesure le niveau de complexité des données de traçabilité en comptant le nombre d'exigences de niveau inférieur ou supérieur qui, pour chaque exigence, a une spécification tracée.

4.4.2.5 Les métriques de validation expérimentation/calculs

Finalement Oberkampff et al. (Oberkampff and Barone 2006) proposent une méthode pour construire deux types de métriques : l'un sur les données expérimentales et l'autre sur le lissage des courbes expérimentales pour mesurer la qualité des modèles numériques ainsi que l'impact des incertitudes expérimentales sur la mesure de la précision.

5 Conclusions

Nous avons présenté, dans ce chapitre, la notion de mécatronique et les différents cycles, méthodes et processus de conception des systèmes mécatroniques. Les spécificités de l'intégration de ces systèmes générant leur complexité ont été précisées. C'est pourquoi, nous nous sommes attachés à montrer la nécessité de trouver un moyen pour évaluer la conception de ces systèmes complexe dans le but d'aider le concepteur à faire des choix.

Certains auteurs se sont penchés sur ce problème et notamment les contributions de Lo (Lo 2013), de Scaravetti (Scaravetti 2004), d'Espagnet (Espagnet 1999a), Zwingmann (Zwingmann 2005) qui ont donné leur définition d'un moyen pour évaluer la conception. Dans ce contexte, la problématique globale peut être formulée de la manière suivante « comment fournir aux concepteurs et architectes, un moyen quantitatif d'évaluer leur conception et notamment les différentes alternatives de conception, pour faciliter la prise de décision ? ». L'approche que nous proposons et développons pour répondre à ce problème se base sur l'emploi des métriques.

Ainsi, après avoir présenté les différents types de métriques, leurs caractéristiques et l'intérêt de leurs usages, une définition des métriques a été formulée, en se basant sur des données de la littérature.

Le domaine dans lequel nous nous positionnons est celui de la conception des systèmes mécatroniques pour lequel peu de travaux existent. En effet, l'état de l'art sur les métriques montre que l'évaluation de la conception se fait souvent sur un domaine en particulier et non sur l'ensemble des disciplines qui composent la mécatronique. De plus, les hypothèses que nous avons identifiées, à savoir qu'une métrique doit être normée, compréhensible, réalisable et sur une échelle d'intervalle adéquate, ne sont pas systématiquement vérifiées. Sans ces conditions, l'évaluation et la comparaison des architectures candidates est difficile. Nous pouvons préciser également qu'aucune de ces métriques n'aborde finement les architectures et les caractéristiques physiques (géométrie, poids, volume) du système qui est un point important en préconception des systèmes mécatroniques.

Les travaux de cette thèse se sont ainsi centrés sur l'évaluation des architectures selon l'objectif d'intégration lors de la conception des systèmes mécatroniques pendant la phase de préconception et notamment pendant la conception préliminaire.

Dans le chapitre suivant, nous nous attacherons à développer, plus précisément des métriques pour le choix d'architecture, en prenant en compte l'objectif d'intégration, fondamental durant la phase de conception des systèmes mécatroniques. Nous proposerons donc notamment des métriques pour évaluer l'intégration des différents domaines de la mécatronique et aborderons l'intégration sur les différents niveaux des architectures de la conception.

Chapitre 2

Métriques d'architectures pour l'intégration multi-domaine et multi-niveau

1	Introduction.....	59
1.1	Le contexte industriel.....	59
1.2	Problématique et plan.....	59
2	Etat de l'art.....	60
2.1	Définitions des architectures	60
2.2	Rôle et besoins de l'architecte système.....	61
2.3	Cycle de vie des architectures.....	63
3	Métriques d'architectures développées	77
3.1	Métrique d'intégration des niveaux d'abstraction	78
3.2	Métrique d'intégration multi-domaine.....	86
3.3	Métriques d'innovation (MI)	94
3.4	Métrique de Mise en Œuvre d'un Projet Mécatronique (MMOPM).....	99
4	Expérimentation.....	101
4.1	Description du système	101
4.2	Résultats	105
4.3	Analyse des résultats et discussions	109
5	Discussions	112
6	Automatisation du calcul des métriques	115
7	Conclusions et perspectives	121

1 Introduction

1.1 Le contexte industriel

Dans l'industrie, beaucoup de concepteurs et d'architectes travaillent sur le même projet, soit à des phases de conception différentes, soit dans des services différents. Ils ont besoin de connaître les raisons du choix des architectures retenues mais également de celles qui ont été écartées. Pour cela, ils doivent souvent mener des recherches d'antériorité longues et fastidieuses, qui, non seulement, ne sont pas adaptées aux cycles de développement actuels de plus en plus courts, mais en plus sans avoir la certitude de trouver finalement ce qu'ils cherchent. La traçabilité des choix de conception est d'autant plus complexe dans la conception mécatronique, que les choix d'architectures sont multi-domaines et font donc intervenir différentes disciplines (O2M 2008; Penas et al. 2009). Ces disciplines n'ont ni les mêmes contraintes, ni les mêmes critères de sélection, et ont parfois même des problèmes d'ontologies, puisqu'initialement elles ne partagent pas le même vocabulaire, les mêmes processus, langages et outils (O2M 2009b). Le projet O2M (Outils de Modélisation Mécatronique), projet collaboratif de 3 ans (2008-2011), dont l'objectif était de développer une nouvelle génération d'outils pour supporter le processus de conception de la mécatronique, a d'ailleurs traité ces aspects en définissant un cahier des charges « type » d'un produit mécatronique pour anticiper ces difficultés (O2M 2009c).

Enfin, un grand nombre de systémiers (automobile, aérospatiale,...) sous-traite la conception de sous-systèmes, de plus en plus complexes, à des entreprises dans différents domaines d'expertise, afin qu'elles leur fournissent des solutions techniques/technologiques pour l'architecture physique de leur système. Elles doivent leur fournir les informations et les modèles correspondant à l'architecture système qu'elles proposent, mais comme plusieurs technologies peuvent répondre à de mêmes exigences initiales, les systémiers ont aussi besoin de disposer de moyens leur permettant de comparer quantitativement les différentes solutions proposées afin de faire le meilleur choix d'architecture et en assurer la traçabilité.

1.2 Problématique et plan

Pour répondre de manière innovante aux attentes en méthodologie des ingénieurs dans l'industrie, il faut leur fournir un outil s'appuyant sur une méthode leur permettant de choisir une architecture répondant au mieux à leurs objectifs, parmi plusieurs architectures candidates, et cela dès les premières phases de conception. Nous aborderons dans ce chapitre l'objectif d'intégration qui est une des caractéristiques majeures des systèmes mécatroniques, et plus particulièrement l'intégration multi-niveau et l'intégration multi-domaine, auxquelles nous avons couplé l'innovation. Pour cela, nous proposons des indicateurs et des métriques d'intégration multi-niveau, multi-domaine, d'innovation pour la conception préliminaire en mécatronique, avant de conclure sur une métrique de mise en œuvre d'un projet mécatronique permettant aux concepteurs et architectes systèmes de choisir l'architecture correspondant au mieux à ces objectifs et aussi de justifier et tracer leurs choix.

Ce chapitre commence par un complément du chapitre 1 sur l'état de l'art en précisant les notions et définitions de base des architectures sur lesquelles ces travaux s'appuient. Il se poursuit avec les méthodes, techniques et outils permettant de générer les architectures et les outils d'aide à la décision permettant de mener à bien leur évaluation dans la phase de

préconception. Puis, la section 3 décrit nos travaux menés sur les métriques d'intégration permettant l'évaluation des différentes architectures solutions notamment pour l'intégration multi-niveau et multi-domaine. Une autre métrique est développée ensuite pour mesurer le degré d'innovation d'un système. Une dernière métrique conclut cette section en couplant ces trois métriques et en créant ainsi une métrique globale pour évaluer quantitativement le coût associé à la mise en œuvre d'un projet mécatronique. Enfin, la quatrième section illustre ces métriques par une expérimentation sur le système mécatronique du moteur-roue. Enfin, nous terminons par présenter nos travaux concernant l'automatisation du calcul de ces métriques.

2 Etat de l'art

2.1 Définitions des architectures

Les définitions générales d'une architecture sont nombreuses et dépendent du système concerné (Benade and Pretorius 2012).

Par exemple, une architecture d'entreprise consiste à mettre en place un cadre d'architecture (cadre Zachman, DoDAF, MODAF, TOGAF...)(Chen et al. 2014) de référence et à aligner les objectifs métiers avec les composantes des systèmes d'information. L'architecture d'entreprise s'intéresse spécifiquement à la modélisation des organisations et des systèmes d'information. Une « entreprise » est définie comme un *ensemble d'individus et de ressources matérielles & logicielles regroupés et organisés autour d'un même objectif, elle fournit une approche commune pour l'intégration des orientations stratégiques, économiques et de gestion de la technologie dans le cadre de la conception de l'organisation et de l'amélioration de la performance.*

Les architectures qui nous intéressent dans ce chapitre sont les architectures de systèmes (Pahl et al. 2007b), dont voici quelques définitions. M. Lo (Lo 2013) définit une architecture comme «une organisation des différents éléments d'un système [...] et des relations entre ces éléments. Cette structure fait suite à un ensemble de décisions stratégiques prises durant la conception... ».

Dans la norme ISO/IEC 42010 :2011 (ISO 2014), « *une architecture est un ensemble de concepts ou de propriétés fondamentaux d'un système dans son environnement représenté par ses éléments, ses relations et par les principes de sa conception et de son évolution* ».

D'après (Rechtin and Maier 2000) « *une architecture est l'organisation fondamentale d'un système représenté par ses composants, leurs relations les uns aux autres et l'environnement ainsi que par ses principes directeurs de sa conception et de son évolution* ».

D'après Faisandier (Faisandier 2013), *une architecture système est une représentation de concepts, et de l'allocation de fonctions physiques/informationnelles à des éléments de forme ainsi que la définition d'interfaces entre les éléments, et avec le contexte environnant. Une architecture système est définie au travers de différentes vues, typiquement une vue fonctionnelle, une vue comportementale, une vue temporelle et une vue physique.*

Enfin, le groupe de travail (SAWG : System Architecture Working Group)¹⁵ sur l'Architecture Système de l'INCOSE (The International Council on Systems Engineering)¹⁶ définit « *une*

¹⁵ <http://www.incose.org/practice/techactivities/wg/sysarch/>

¹⁶ <http://www.incose.org/>

architecture système comme une structure fondamentale et unificatrice du système, définie en termes d'éléments du système, d'interfaces, de processus, de contraintes et de comportements ».

Dans le processus d'ingénierie système, la conception architecturale consiste à synthétiser une solution du système qui satisfasse les exigences (Figure 2-1), en formalisant plusieurs niveaux d'architectures.

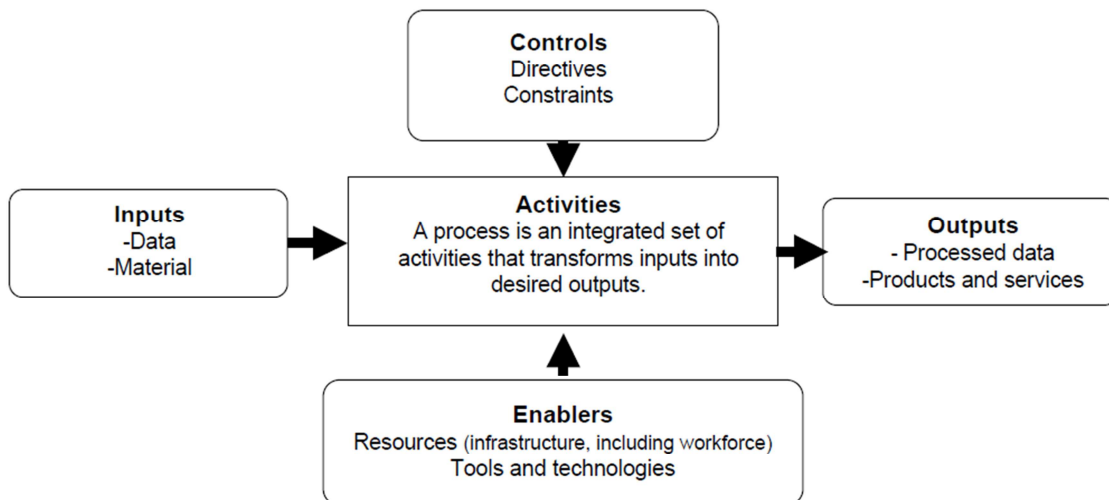


Figure 2-1 : Processus de conception architecturale vu par l'INCOSE (INCOSE 2006)

Les différents niveaux d'architectures (fonctionnelle, logique et physique) ont été détaillés au Chapitre 1 §1.2.3.2.

2.2 Rôle et besoins de l'architecte système

Développer l'architecture système est la tâche principale de l'architecte système. Il a pour objectif de concevoir l'architecture du système, préparer les scénarii de simulation, et valider l'architecture à partir de simulations effectuées. Pour cela, l'architecte système doit comprendre les besoins et les attentes du client mais aussi des fournisseurs, des collaborateurs (sous-traitants)...

L'architecte système a une variété d'activités (Figure 2-2) principalement dans la collecte, le filtrage, le traitement des données mais aussi des activités plus formelles comme des réunions, des visites... La majorité de son temps sert à la communication entre les différents membres du projet. Il est en contact avec plusieurs interlocuteurs comme le directeur commercial, le chef de projet, le directeur de marketing, le responsable technique, les ingénieurs, les concepteurs... Il doit avoir des connaissances détaillées mais aussi savoir prendre du recul, pour pouvoir passer rapidement de vues détaillées spécifiques à des vues

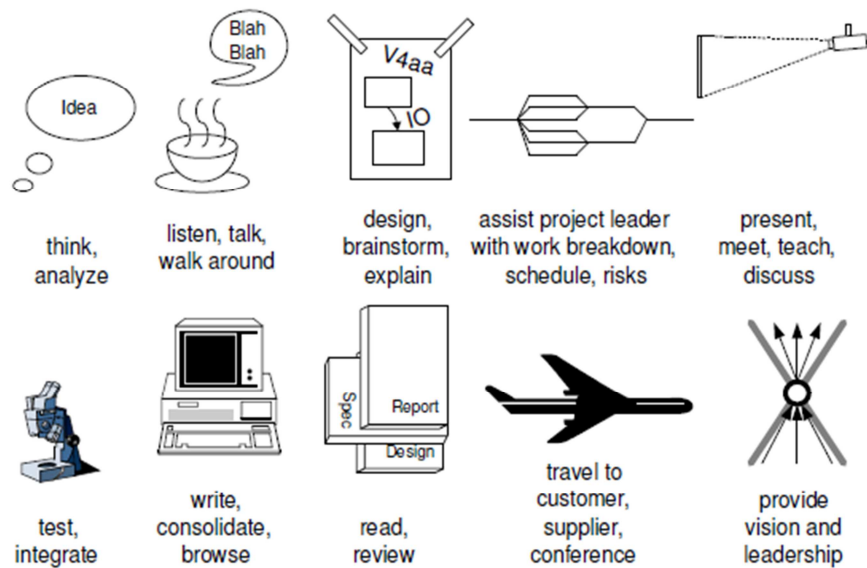


Figure 2-2 : Diversité des activités de l'architecte système (Muller 2011)

Il doit également saisir les enjeux du produit et du cadre d'architecture qu'on lui a fixé, afin de pouvoir étudier les différents choix de conception et les solutions techniques et prendre les bonnes décisions en fonction d'un certain nombre de critères (maturité de la solution, prix, risques, bénéfices...). Il a une vision et une compréhension d'ensemble du système et de son contexte. L'architecte système doit permettre aux membres des équipes spécialisées (qui ont un champ de vision limité) de prendre des décisions de conception locale en leur fournissant des informations de conception globale. Ainsi, il définit d'abord le concept, fixe des objectifs de performance, réalise l'analyse fonctionnelle, et pour finir conçoit les architectures. C'est un processus créatif et il peut y avoir beaucoup de variantes possibles d'architectures qui satisfassent les exigences. L'architecte système a donc besoin de méthodes et d'outils pour analyser ces architectures et choisir celle qui répondra au mieux aux objectifs de conception haut-niveau fixés. La décomposition des systèmes en sous-systèmes, des sous-systèmes en modules etc. est une responsabilité majeure de l'architecte système. En effet, cette décomposition doit permettre la compréhension des systèmes complexes, tout en veillant à faciliter l'intégration physique de ce système. En effet, l'architecte doit assurer en permanence la cohérence de la conception tout au long de la phase de conception préliminaire (du niveau système au niveau physique)(Muller 2011).

Ce travail de compromis vis-à-vis des objectifs de conception ou des exigences qui peuvent être parfois contradictoires, constitue une des activités principales de la conception préliminaire. L'architecte système est garant de l'équilibre et des compromis (par exemple le coût par rapport à la valeur ajoutée en termes de fonctionnalités et de performances du système). Il est en charge de l'intégrité des spécifications du système au fil de la conception. Si un paramètre évolue, l'architecte système doit vérifier et garantir que les autres paramètres de conception répondent toujours au cahier des charges et conservent les performances.

Aussi, le travail de l'architecte système s'avère critique pendant la conception préliminaire, car, une fois l'architecture choisie suivant les objectifs « haut-niveau », elle constituera la base du travail de toutes les équipes de conception/développement (INCOSE 2006).

2.3 Cycle de vie des architectures

La conception architecturale consiste à décomposer itérativement le système en sous-systèmes, en allouant et en vérifiant les exigences et à spécifier les besoins des sous-systèmes feuilles (modules, pièce) de l'arborescence technique ainsi constituée, ainsi que les interfaces. Même si certains langages et outils sont plus compatibles à l'approche d'IS dans ses concepts globaux (nous citerons par exemple le langage SysML), d'autres outils ou approches permettent de remplir seulement quelques phases du cycle de vie des architectures, sans nécessairement assurer la cohérence entre les différents niveaux d'architecture, ni formaliser la traçabilité de ces architectures avec les exigences... Néanmoins, suivant la structure, la culture d'entreprise et l'organisation du processus de conception, toutes ces méthodes sont couramment utilisées, chacune avec ses avantages et ses inconvénients.

2.3.1 Recherche préalable de concepts / Analyse fonctionnelle

La recherche de concept débute par la conception architecturale après l'analyse du besoin. Cette étape souvent appelée analyse fonctionnelle externe, permet de formaliser les exigences et l'architecture fonctionnelle pour la construction du cahier des charges fonctionnel. Généralement, il y a plusieurs approches pour initier la conception d'un produit : à partir soit d'une page blanche pour la recherche d'une idée innovante, soit d'un système existant que l'entreprise souhaite faire évoluer, en lui rajoutant des fonctionnalités ou de nouvelles technologies, en modifiant son aspect visuel ou en modifiant ses performances.

Nous avons identifié deux méthodes, qui sont développées en *Annexe 1*, pour réaliser cette analyse fonctionnelle externe. La première appelée brainstorming est une méthode de créativité utilisée en assemblée restreinte et de façon informelle pour faire émerger les concepts. La deuxième, APTE (APplication des Techniques d'Entreprise) (O2M 2009c) qui est standardisée par la norme NF X50-100 (NF 2011) permet de définir les frontières du système et fournit à l'aide d'une technique graphique le support nécessaire pour générer un cahier des charges.

Nous citerons également les méthodes basées sur l'approche MBSE et souvent implémentées en langage SysML qui a été présentée dans le Chapitre 1 §1.2.2.2 (Mhenni et al. 2014).

2.3.2 Recherche et émergence d'architectures candidates

La première étape de la conception architecturale consiste en la définition de la structure répondant aux fonctions décrites précédemment. Il s'agit de rassembler des connaissances pour faire émerger plusieurs solutions candidates. Cette étape repose aussi sur des notions plus abstraites et plus difficiles à formaliser telles que la créativité ou l'expérience. Le premier choix est fait, en fonction de la satisfaction plus ou moins complète du cahier des charges (Espanet 1999a).

Nous citerons deux méthodes : la méthode TRIZ et la méthode basée sur le langage SysML précédemment vue au Chapitre 1 §1.2.2.2. Les nombreuses méthodologies construites sur le langage SysML (Verries 2010) permettent également de formaliser les architectures dans la phase d'analyse interne. Les diagrammes utilisés peuvent être les diagrammes de structure de « block definition diagram » (bdd) offrant une vue statique des différents niveaux d'architectures, intégrant les liens d'allocations entre les éléments et de traçabilité entre les différents niveaux d'architecture et les exigences. Toutefois, les diagrammes d'activités (act)

pour l'architecture fonctionnelle et internes de blocs (ibd) pour les architectures logiques et physiques peuvent être utilisés pour avoir une vision plus « dynamique » des architectures, en présentant les flux d'entrées/sorties et leur nature entre chaque entité (Mhenni et al. 2014).

La méthode TRIZ, détaillée dans l'Annexe 2, est une approche algorithmique pour résoudre les problèmes techniques sur les prémices d'une base de connaissances (Figure 2-3).

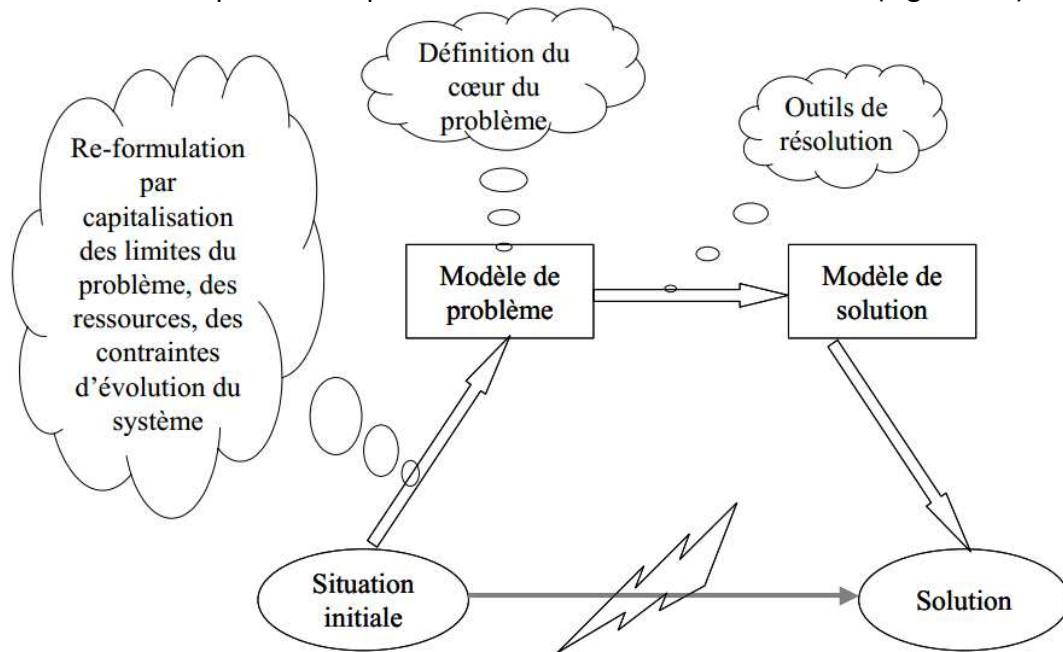


Figure 2-3 : Démarche générale de la méthode TRIZ (Dubois 2004)

2.3.3 Formalisation/modélisation des architectures

Les architectures du système peuvent être modélisées par différents langages et outils suivant leur nature (fonctionnelle, logique, organique ou physique). Parmi ces langages et outils, nous citerons ceux qui actuellement sont les plus utilisés par une grande partie de la communauté mécatronique : la méthode SADT, la méthode FAST, les outils de modélisation et de simulation comme Simulink, Simplorer ou et les langages comme Modelica ou VHDL (détaillés dans l'Annexe 3).

La modélisation SADT (Structured Analysis and Design Technique) est une méthode d'analyse et de conception fonctionnelle descendante par niveaux successifs décrivant le système (Figure 2-4)^{17,18,19}, qui a été complétée par les méthodes et langage les méthodes/langages IDEF (ICAM DEfinition) pour améliorer la productivité en fabrication, qui sont devenues maintenant des standards (FIPS 1993a; FIPS 1993b; IEEE 1998).

¹⁷ <http://adrcomptefreeflo.free.fr/Nouveau%20dossier/IDEF-SADT.pdf>

¹⁸ <http://www.idef.com/>

¹⁹ <http://en.wikipedia.org/wiki/IDEF>

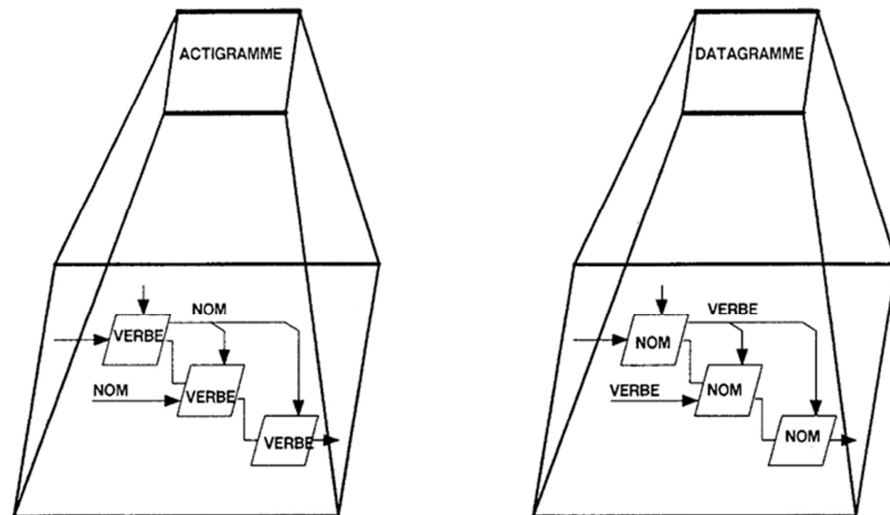


Figure 2-4 : Architecture fonctionnelle (actigramme) et organique (datagramme) avec SADT/IDEF⁵¹

La technique FAST²⁰ (Functional Analysis System Technique), développée en 1964 par C. Bytheway, maintenant décrite dans la norme NF EN 12973 (NF 2000)], permet de représenter, de gauche à droite, les fonctions de services traduites en fonctions techniques puis en solutions techniques. Elle peut donc adresser tous les niveaux d'architecture, de fonctionnelle à physique, même si généralement les solutions techniques sont plutôt des organes que des composants strictement définis (Figure 2-5).

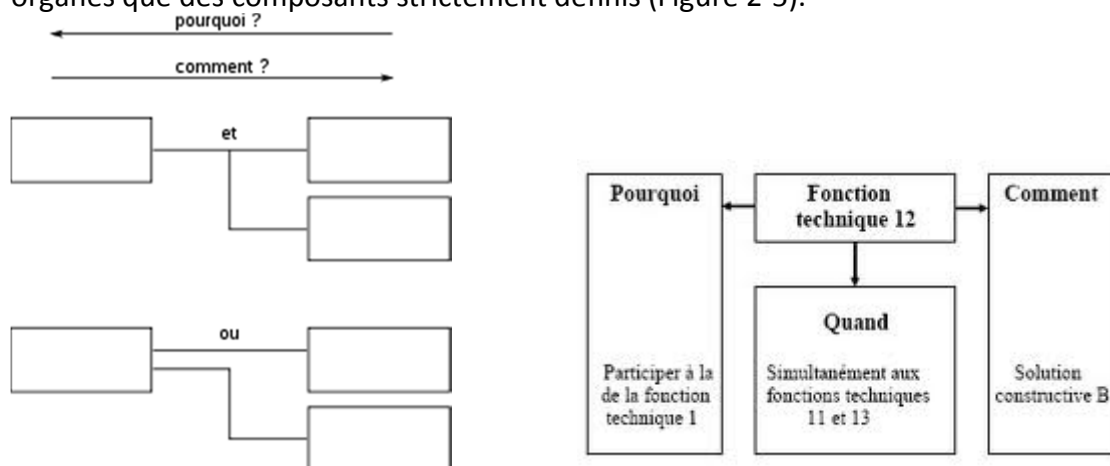


Figure 2-5: Principe des architectures FAST²¹

Les avantages de cette technique est qu'elle permet de modéliser des architectures avec des fonctions logiques comme le « et » et le « ou » ; qu'elle offre une conception créative en proposant des alternatives de réponse pour chaque fonction. Le modèle FAST constitue un ensemble de données permettant une bonne connaissance du système. Enfin, cet outil est normalisé en France (norme NF EN 1325-1(NF 2014)) et permet donc une diffusion assez large.

Les architectures logiques définissent la structure des entités logique qui répondent aux exigences du système : la performance, l'évolution dans le temps, les flux d'énergies, les données échangées, etc. » (Hammadi 2012a) en modélisant le comportement du système à l'aide d'un ensemble d'équations permettant ainsi la simulation (§2.3.4.6). L'architecture physique définit alors les éléments physiques correspondant à ces entités logiques. En effet,

²⁰ http://www.value-eng.org/pdf_docs/monographs/FAbasics.pdf

²¹ <http://intratek.legtux.org/Intra-Tek/Pages/3/4-Projet3-Maison/Pages/3-Fast/index.htm>

En ingénierie système, le niveau physique est le niveau où les composants sont choisis et caractérisés. Les architectures physiques correspondantes peuvent être modélisées avec les mêmes outils que pour l'architecture organique. Pour l'architecture physique, le concepteur peut également s'appuyer sur la modélisation hiérarchique (Figure 2-6) présentée dans l'Annexe 4.

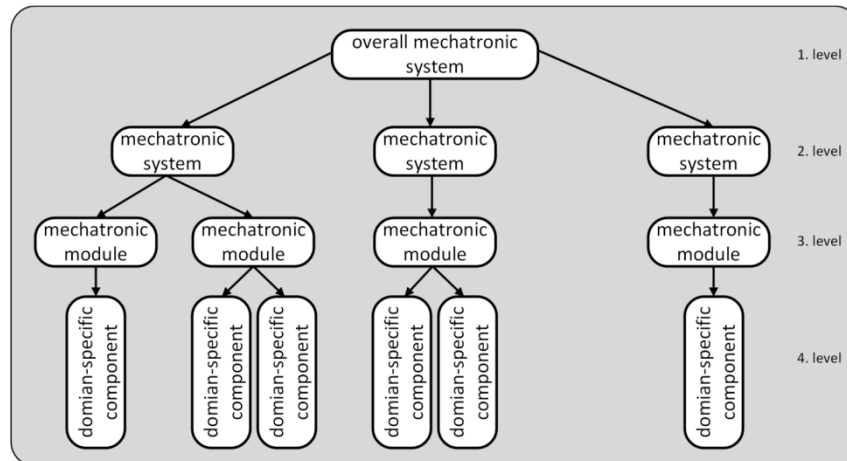


Figure 2-6 : Structure hiérarchique d'un système mécatronique (Kellner and Hehenberger 2014)

Cependant, lorsque les composants (mécanique, électronique etc.) et donc leur géométrie sont connus, il devient intéressant d'utiliser des logiciels de CAO (Conception Assistée par Ordinateur) pour agencer ces différents composants dans l'espace, afin d'évaluer les architectures spatiales candidates du système (voir Chapitre 3). En mécanique, par exemple, l'architecture physique du système technique est matérialisée par des modèles géométriques 2D ou 3D des composants du système qui réalisent la maquette 3D. Les logiciels utilisés à cet effet sont généralement les logiciels de conception assistée par ordinateur(CAO), par exemple CATIA²² et SolidWorks²³ de Dassault Systèmes, FreeCad²⁴ (outil libre), TopSolid²⁵ de Missler Software, Creo²⁶ de PTC, Nx²⁷ de de Siemens, etc. En électronique, il existe également des logiciels de CAO permettant de placer les composants sur différents support (PCB, Flex,...) : Eagle²⁸, Orcad²⁹, Cadence³⁰ etc. Il existe aussi quelques logiciels de CAO pour les domaines de l'électrotechnique (Ecodial), de l'électromagnétisme (Inca3D), ... mais moins courants.

Enfin, notons que ces logiciels de CAO peuvent également être déjà utilisés au niveau logique avec les volumes d'encombrement attribués pour les organes, afin de formaliser en phase amont les premières contraintes géométriques/volumique du cahier des charges.

2.3.4 Evaluation / Analyse des architectures

Pendant la conception préliminaire, les fonctionnalités sont dérivées soit du cahier des charges ou soit de l'analyse fonctionnelle. Plusieurs solutions techniques sont possibles pour répondre à ces fonctions et des choix doivent donc être analysés pour passer aux phases

²² <http://www.3ds.com/fr/produits-et-services/catia/>

²³ <http://www.solidworks.fr/>

²⁴ <http://www.freecadweb.org/>

²⁵ <http://www.topsolid.fr/>

²⁶ <http://fr.ptc.com/product/creo>

²⁷ http://www.plm.automation.siemens.com/fr_fr/products/nx/

²⁸ <http://www.cadsoftusa.com/>

²⁹ <http://www.orcad.com/>

³⁰ <http://www.cadence.com/us/Pages/default.aspx>

suivantes : choix d'architecture, choix de solutions et de composants physiques. Cette phase du processus de développement, essentiellement consacrée à étudier les différentes solutions d'architectures selon plusieurs objectifs comme la performance, les coûts, la maintenabilité, la robustesse,... conduisent à des décisions pouvant fortement influencer le coût, la réussite, le temps de développement d'un projet.

2.3.4.1 Cadre et définitions

Lo (Lo 2013) indique que « l'évaluation en conception consiste à fournir des éléments factuels, rationnels, permettant de comparer des alternatives de solutions ou solutions candidates et de juger de leur adéquation avec les exigences issues des besoins des parties prenantes ». Le processus de décision est décrit par (Ullman 2000) dans la Figure 2-7.

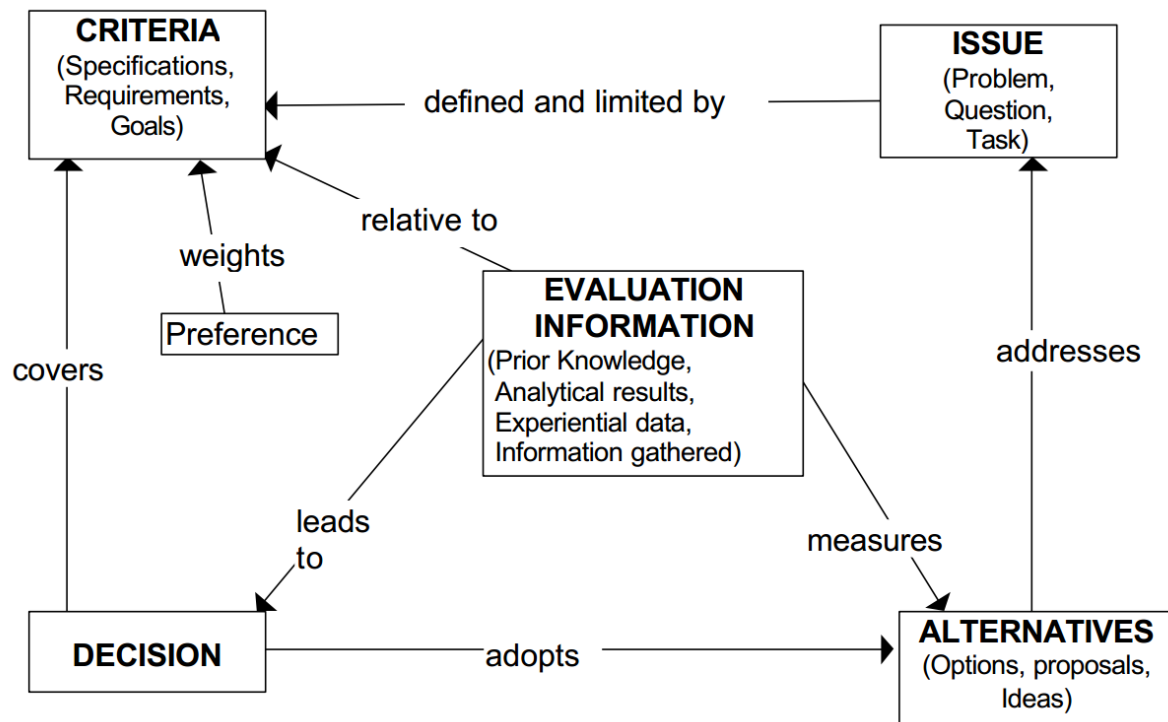


Figure 2-7 : Processus d'aide à la décision (Ullman 2000)

Un *problème* est délimité et défini par des critères.

Un *critère* est une règle de jugement qui sert à juger du niveau de satisfaction des *alternatives*. Cette règle est définie sur l'ensemble des actions représentant les préférences de l'utilisateur, selon son point de vue. Le critère *peut être pondéré* en fonction des souhaits des parties prenantes. Le poids mesure l'importance d'un critère par rapport aux autres, du point de vue du *décideur*. Les *informations d'évaluation* sont les connaissances disponibles sur les solutions existantes, les résultats d'analyse, les données expérimentales,... Les décideurs prennent alors une *décision* en fonction des résultats de l'évaluation.

Dans le paragraphe suivant, nous allons détailler les différents outils qui s'offrent aux concepteurs dans l'analyse des architectures, préalable à la prise de décision pour le choix de ces architectures.

2.3.4.2 Quality Function Deployment (QFD) – Maison de la qualité

Le principe remonte à la fin des années 1960 et a été développé au Japon par Yoji Akao en 1966. QFD est un concept global d'analyse pour l'aide à la décision dans la conception de

produits ou de services. Cette approche est basée sur les matrices et permet une meilleure prise en compte de la perception du client par rapport aux avantages et au coût du produit. Les matrices permettent de représenter les caractéristiques et paramètres critiques pour le client (attentes, besoin du marché, désirs des futurs utilisateurs) recensés par exemple lors d'enquêtes marketing ou d'analyses comparatives, et de les coupler aux différentes solutions envisagées. La grille réalisée, surnommée « Maison de la qualité » (en anglais : « House of Quality », HoQ) est un outil de la méthode QFD, apparu en 1972 chez Mitsubishi Heavy Industries, qui permet ainsi d'évaluer les meilleures solutions vis-à-vis du client dès la conception du produit ou du service.

Cette approche globale vise à une meilleure prise en compte des attentes/avis du client dans toutes les étapes du processus de conception du produit.

Cette technique peut être également appliquée aux choix d'architectures, des sous-systèmes, d'assemblages, en remplaçant les « caractéristiques techniques » par l'élément idoine, suivant le « numéro » de la HoQ :

La première « maison » (HoQ I) est consacrée à l'« Organisation Produit » et est montrée dans la Figure 2-8.

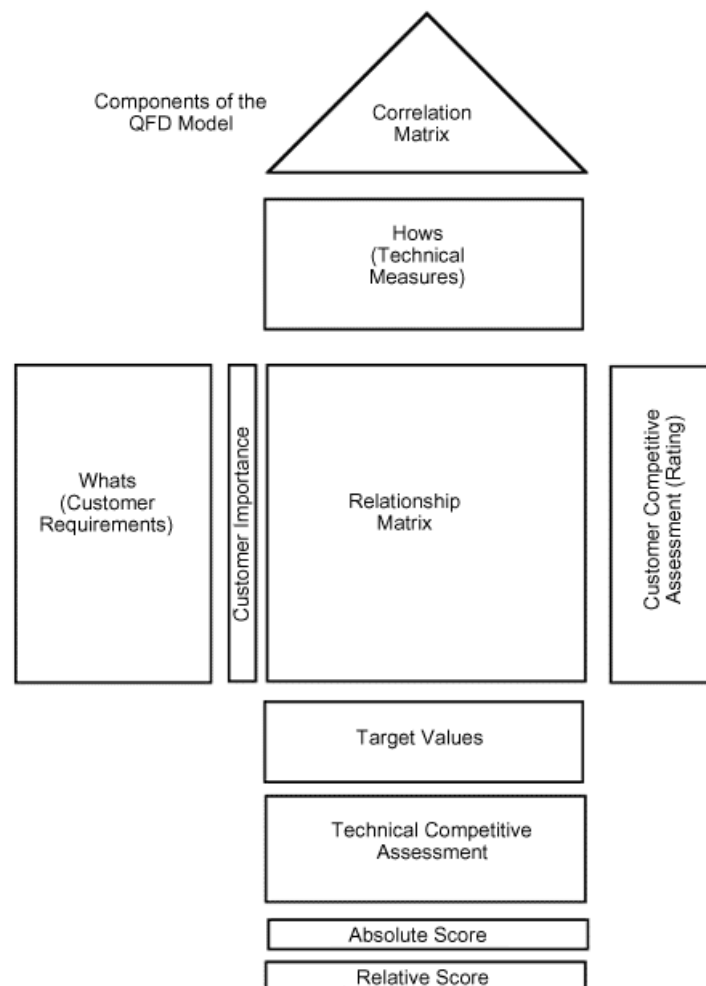


Figure 2-8 : Maison de la qualité HoQ I

Elle a pour but de traduire les souhaits des clients et leurs priorités dans une description du produit de manière technique et quantifiable. Et ainsi les priorités des caractéristiques techniques peuvent être dérivées.

La construction de cette « maison de la qualité » (HoQ I) nécessite plusieurs étapes :

1. Les *besoins des clients* sont identifiés. Cette étape est cruciale pour la réussite du produit et doit être accompli avec beaucoup d'attention.
2. Les besoins clients sont structurés et *priorisés* (par exemple entre 1 et 10 ou 1 et 100)
3. La *satisfaction du client* est analysée par rapport à la réalisation de ses besoins. De plus, lorsque le produit existe, il est comparé à ceux des concurrents.
4. Les *caractéristiques techniques* sont identifiées. Dans cette étape, l'équipe QFD essaye de traduire les besoins du client en caractéristiques techniques.
5. Une *direction d'optimisation* des caractéristiques techniques est déterminée, c'est-à-dire la direction dans laquelle les caractéristiques techniques devraient amener une perception améliorée des bénéfices pour le client.
6. La *matrice de corrélation* permet de déterminer les relations entre les différentes caractéristiques techniques. Elle permet d'évaluer le degré d'interrelation entre les caractéristiques techniques du produit, c'est-à-dire si deux caractéristiques rentrent en conflit ou au contraire sont redondantes. De plus, elle met en évidence les liens de communication nécessaires entre les différentes équipes qui travaillent sur le même projet.
7. La *matrice de relations* définit, comme son nom l'indique, les relations entre les besoins clients et les caractéristiques techniques. Les caractéristiques techniques sont évaluées afin de connaître à quel degré elles satisfont les besoins client. C'est un jugement qualitatif. Ces caractéristiques sont alors priorisées.
8. L'étape 8 est semblable à l'étape 3. Cette fois-ci les caractéristiques techniques sont comparées à la concurrence.
9. L'importance des caractéristiques techniques est calculée : le poids de ces caractéristiques est égal à la somme des poids des besoins client multipliée par le taux de satisfaction des besoins de cette caractéristique.
10. Pour finir, les valeurs cibles des caractéristiques techniques sont déterminées.

La première « maison » HoQ I correspond à l'organisation du produit, c'est-à-dire la traduction des besoins « client » en caractéristiques techniques, mais une deuxième « maison » peut être construite en partant des éléments produits par la première. Les informations d'un niveau supérieur sont transférées à un niveau inférieur. Les étapes 4 et 5 et l'étape 10 de la première maison deviennent l'étape 1 de la seconde. La sortie d'une matrice est l'entrée de la suivante.

Ainsi, il en résulte quatre « maisons » (Figure 2-9) :

- HoQ I : Organisation du produit (besoins client, caractéristiques techniques)
- HoQ II : Déploiement des composants (caractéristiques techniques, caractéristiques des composants)
- HoQ III : Planification des processus (caractéristiques des composants, besoins processus)
- HoQ IV : Planification de la production (besoins processus, besoins de production)

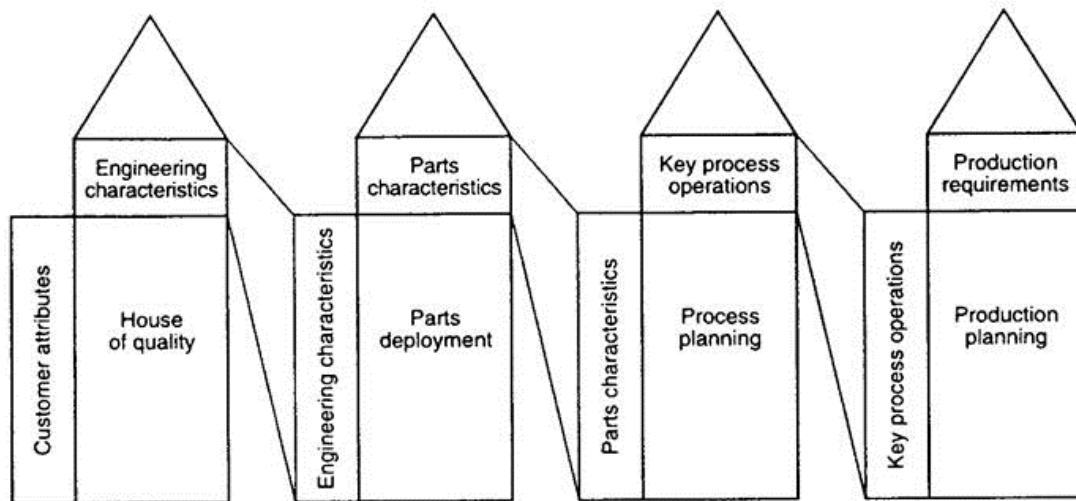


Figure 2-9 : Les quatre maisons de la qualité (Towar 2008)

Cette méthode prend en compte les besoins et les priorités des clients, la probabilité de succès sur le marché est augmentée. De plus, cet outil encourage la coopération des différentes équipes techniques, dont la synergie ne peut qu'être favorable.

2.3.4.3 DSM (Design Structure Matrix) – DMM (Domain Mapping Matrix)³¹

Les DSM (Design Structure Matrix ou encore Dependency Structure Matrix) ont été créées par Steward en 1981. L'équipe DSM Group at MIT (DSM) travaille à développer la théorie des DSM. Les DSM sont des matrices carrées ayant pour lignes et colonnes les composants d'un système. L'élément (i, j) de la matrice représente une relation entre le composant de la ligne i avec celui de la colonne j (Figure 2-10).

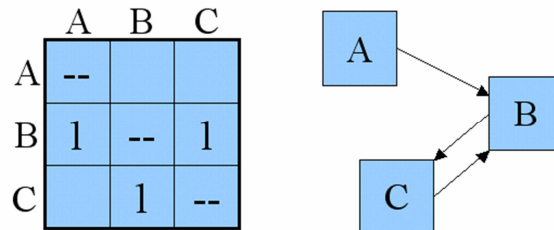


Figure 2-10: Exemple de DSM.

Une DSM est donc une image matricielle d'un graphe qui représente le système. C'est une méthode qui permet de présenter l'architecture d'un système complexe en sous-systèmes moins complexes. Les DSM représentent les interactions entre des éléments d'un même domaine (exemple entre différents composants, différentes fonctions,...), tandis que les DMM (Domain Mapping Matrix) décrivent les interactions/dépendances entre deux domaines distincts. Les interdépendances des DSM/DMM entre les sous-systèmes peuvent être de différents types (Turki 2008a):

- Les DSM basées sur les composants : elles peuvent exposer plusieurs types d'interactions (spatiales, énergétiques, informationnelles, échanges de matière, etc.).
- Les DSM dédiées aux équipes de travail : elles sont utilisées pour les analyses organisationnelles et les flux informationnels entre individus ou groupes participant à un projet.

³¹ <http://www.dsmweb.org/>

- Les DSM d'activités : elles expriment les interdépendances entre les tâches d'un certain processus.
- Les DSM paramétrées : elles sont construites en découpant un système, en définissant les interactions entre ses composants puis en quantifiant un type d'interactions (proximité spatiale, liaison mécanique, besoin de transferts énergétiques, etc.).

Les DMM peuvent ainsi établir une cartographie entre deux éléments de niveau d'architecture différent d'un système (Figure 2-11). Ainsi, en ce qui nous concerne les architectures, la cartographie entre les fonctions et les composants (Figure 2-12 « Functionality/Product »), ou les organes et les composants physiques est celle qui nous intéressera le plus.

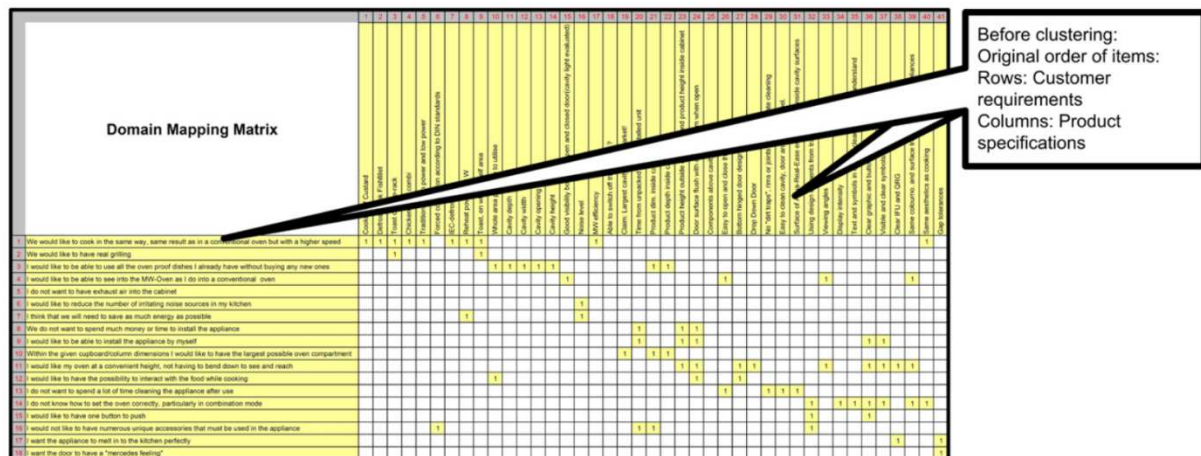


Figure 2-11 : Exemple d'une représentation DMM entre les exigences clients et le cahier des charges du produit (Danilovic and Browning 2007)

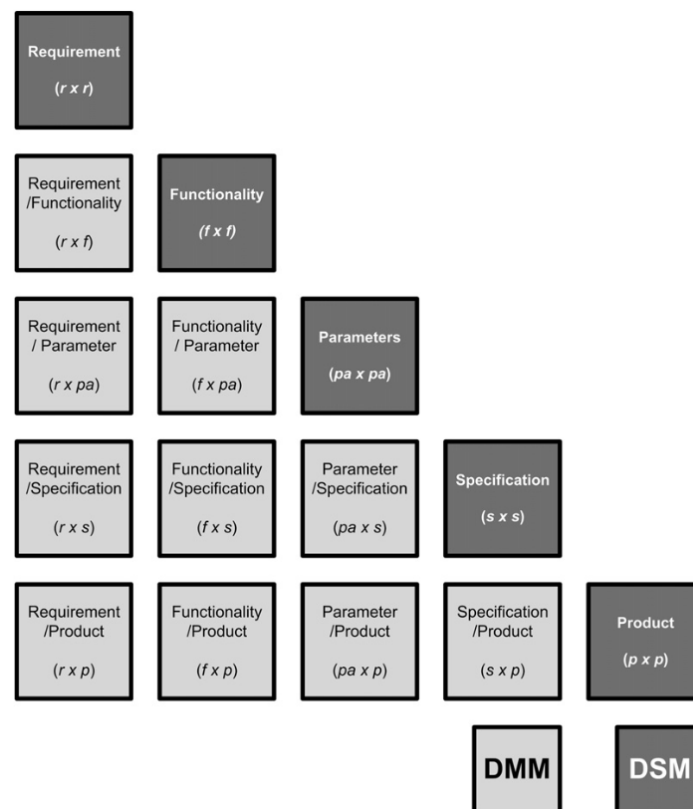


Figure 2-12 : DSM et DMM spécifiques pour le système et le produit (Danilovic and Browning 2007)

Cette représentation de l'architecture d'un système sous forme d'une matrice a pour avantage d'offrir une structure de données bien adaptée aux manipulations algorithmiques (transpositions de lignes et de colonnes, parcours sur les éléments etc.) et aussi aux manipulations mathématiques (additions, multiplications de matrices, recherche de valeurs propres etc.) (Turki 2008a).

Les matrices utilisées dans les matrices QFD précédentes ou dans la conception axiomatique ci-après sont des DMM particulières.

2.3.4.4 Axiomatic design

L'Axiomatic Design est une approche d'évaluation inventée par Suh (Suh 1990), basée sur un objectif de réduction de la complexité de la conception. Elle évalue l'adéquation d'une architecture aux besoins du client vis-à-vis de son respect de règles appelées axiomes. Selon Suh, la conception suit un cheminement itératif et séquentiel entre "ce qu'on cherche à accomplir" et "comment l'accomplir". Il définit la conception comme optimale lorsque elle est composée de deux axiomes : le premier est l'indépendance des exigences fonctionnelles (FR) et le deuxième est de minimiser le contenu de l'information.

Le processus de conception est divisé en quatre domaines (Figure 2-13) :

- *le domaine client* est caractérisé par les besoins du client, désignés par {CA} pour « Customer Attributes », correspondant au cahier des charges.
- *le domaine fonctionnel*, dans lequel les CAs sont traduits par des fonctions à atteindre ({FR} : « Functional Requirements »). Ces exigences fonctionnelles sont définies comme le nombre minimal d'exigences qui caractérisent complètement le besoin fonctionnel (identifié dans l'architecture fonctionnelle).
- *le domaine physique* permet d'identifier ou de sélectionner les paramètres physiques ou les paramètres de conception {DP} (« Design Parameters »), qui caractérise chacun une solution répondant à un FR. Selon le raffinement de la décomposition, le domaine physique peut correspondre soit à l'architecture logique, soit à l'architecture organique ou physique.
- *le domaine du processus* permet d'exprimer les variables des processus {PV} pour « Process Variables » qui sont choisies à partir des DPs.

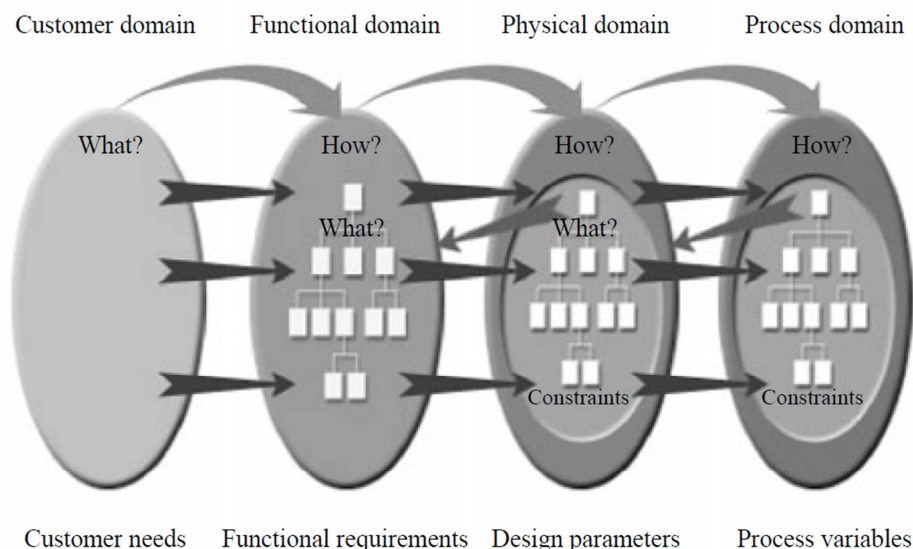


Figure 2-13 : Les quatre domaines de l'Axiomatic Design (Park 2007)

Pour évaluer le premier axiome, une matrice de conception [A] est construite et indique le couplage entre les exigences fonctionnelles et les solutions DP :

$$[FR] = [A][DP] \quad (2.1)$$

Trois cas peuvent se présenter :

- Une matrice [A] diagonale signifie l'absence de couplage fonctionnel. Chaque FR peut être satisfait par un DP indépendant.
- Une matrice triangulaire caractérise une conception découplée au sens où les DP peuvent être utilisés séquentiellement pour satisfaire les exigences fonctionnelles.
- Les autres types de matrices correspondent à des couplages qu'il faut éviter pour respecter l'axiome 1. Dans ce cas, la conception est à refaire.

Le deuxième axiome est celui du minimum d'informations contenues dans la conception. Comme le contenu de l'information peut être défini différemment selon les caractéristiques de la conception, comment pouvons-nous définir quantitativement la mesure de l'information? La définition varie ainsi en fonction de la situation. En général, l'information est liée à la complexité. Ainsi comme introduit au Chapitre 1 §4.2, Suh propose de considérer un indice de la teneur en information relativement à la probabilité de succès de répondre aux exigences fonctionnelles FRs. Pour chaque exigence fonctionnelle (FR_i) un « contenu » d'information I_i peut être défini par l'équation :

$$I_i = \log_2 \left(\frac{1}{P_i} \right) = -\log_2(P_i) \quad (2.2)$$

Avec P_i la probabilité de satisfaire FR_i.

Concernant les architectures optimales pour la mécanique, cette approche est un peu ambiguë, notamment d'un point de vue de l'intégration physique : si un design simple peut apparaître en première approche comme une bonne chose, nous pourrions penser qu'une bonne conception consisterait à trouver un DP répondant à plusieurs FRs, formalisant une conception « couplée ». Cependant, du point de vue de l'approche axiomatique, si l'entité physique se compose de plusieurs DPs, cette intégration physique peut être recommandée, si elle maintient l'indépendance des FRs, puisqu'elle réduit la complexité (la quantité d'informations du deuxième axiome) de la conception (Park 2007).

Cette méthode est utilisable pour évaluer la complexité d'une architecture, lorsque l'architecture logique avec les équations paramétriques liées aux exigences de performances ou de comportement ont permis de formaliser les composants organiques (technologies) (DPs) correspondants. Ainsi, elle sera plutôt utilisée en phase de pré-dimensionnement pour valider une architecture, une fois que les technologies/ organes auront été choisis, au même titre que les outils de simulation « logique ».

2.3.4.5 Prise en compte des incertitudes

Les incertitudes peuvent avoir des origines différentes : sur le modèle ou sur les paramètres, incertitudes numériques (Zerelli 2014)... Or, même si la phase de préconception est une phase de créativité, bien souvent, le concepteur s'oriente parfois directement vers une solution technique qu'il connaît, et donc des solutions intéressantes peuvent être mises de côté parce qu'elles ne sont pas connues, non habituelles ou mal estimées (Scaravetti 2004). Pour cela, un critère de confiance peut être ajouté pour évaluer le risque de conception vu par le concepteur (Moullec et al. 2012b). La distance à une solution de référence, qui est

connue et bien cotée par l'entreprise, contribue à élaborer cet indice de confiance (Pailhes et al. 2011).

Les architectures peuvent être enrichies à l'aide de Réseaux Bayésiens (Bayesian Networks BN), pour analyser la prise de risques correspondante pour chaque architecture.

Un réseau bayésien est un modèle probabiliste graphique qui représente un ensemble de variables aléatoires et leur distribution de probabilité conjointe. C'est un graphe acyclique orienté dont les nœuds représentent les variables et les arcs représentent les dépendances conditionnelles. Les réseaux bayésien sont utilisés afin de générer l'ensemble des architectures de produits possibles en tenant compte des connaissances incertaines des concepteurs. Un réseau bayésien est donc à la fois un modèle de représentation des connaissances et une machine à calculer les probabilités conditionnelles.

Les deux principales étapes sont les suivantes :

1. définir le graphe du modèle ;
2. définir les tables de probabilité de chaque variable, conditionnellement à ses causes.

En fonction des informations observées, on calcule la probabilité des données non observées.

Moullec et al. (Moullec et al. 2012b) utilisent les réseaux bayésiens couplés à la méthode CSP (Constraint Satisfaction Problem, voir paragraphe suivant) pour générer l'ensemble des architectures de produits possibles en tenant compte de la connaissance incertaine des concepteurs et les évaluer en s'appuyant sur l'optimisation du placement des composants (Moullec et al. 2013). Pour cela, ils représentent un ensemble de contraintes déterministes qui peuvent être ou non remplies par des architectures candidates. Leur méthode comporte deux étapes (Moullec et al. 2012b). La première est manuelle et consiste à construire un réseau Bayésien, en s'appuyant sur 6 «formats» :

- un pour définir les différentes solutions candidate,
- un pour les données relatives aux caractéristiques des composants définissant les variables de décision et utiles pour l'évaluation des performances,
- un pour les performances à atteindre pour chaque architecture et le seuil associé à respecter,
- un pour les contraintes que l'architecture doit satisfaire,
- un pour évaluer les incertitudes de la compatibilité de chaque composant avec les autres.
- un pour la faisabilité de l'architecture globale qui représente le critère de confiance de la solution.

La deuxième étape est entièrement automatisée et consiste en la génération par un programme de l'ensemble des solutions admissibles, basée sur l'inférence bayésienne. Seules les architectures qui répondent à toutes les contraintes de performance définies et au niveau de confiance choisi seront générées.

Concernant le couplage avec l'optimisation du placement des composants (architecture spatiale), leurs travaux sont détaillés ci-dessous dans la partie « Architecture physique (spatiale).

2.3.4.6 Evaluation des architectures logiques

Les outils de modélisation pour l'architecture logique, présentés précédemment (§2.3.3) permettent également de simuler ces différentes architectures candidates pour vérifier lesquelles présentent les performances attendues. Ainsi, les résultats de simulation permettent d'avoir une cartographie des architectures qui permettent de répondre aux

exigences, avec en plus la ou les valeurs admissibles des paramètres de conception pour que l'architecture évaluée reste valide.

2.3.4.7 Analyse de l'architecture physique (spatiale)

L'analyse de l'architecture spatiale des composants physiques sera traitée plus en détail dans le chapitre 3. Toutefois, on peut déjà citer les travaux de Moullec et al. (Moullec et al. 2013) qui consistent à ajouter à leur méthode de couplage Réseaux Bayésien / CSP, une contrainte dimensionnelle, permettant de définir la quantité/dimension nécessaire d'un composant si on l'associe à un autre ou de coupler les performances requises à la quantité / dimension d'un composant. De plus, pour chaque composant sont définis sa largeur, sa hauteur et sa profondeur. Les variables de décision associées sont les coordonnées et les 6 orientations possibles. Enfin, cinq nouvelles contraintes ont été créées : la contrainte de position, la contrainte d'adjacence, la contrainte d'orientation, une contrainte de distance, et une contrainte de performance relative à l'assemblage (et aux positions des composants). Ces travaux permettent de déterminer si pour une architecture donnée, avec des dimensions et formes de composants fixés (se limitant à des composants de forme parallélépipédique) la position et orientation de optimale de chaque composant permettant d'atteindre la performance attendue.

2.3.5 Aide à la décision pour le choix des architectures

Pour concevoir un système optimal, le concepteur doit, après avoir exploré toutes les alternatives générées par les différentes solutions, les comparer pour finalement faire ses choix de conception (Scaravetti 2004). C'est pourquoi il est important d'aider les concepteurs à faire leur choix d'architectures, tout en leur donnant la possibilité de capitaliser la traçabilité de leurs choix de conception (Lago et al. 2008). Les outils d'aide à la décision permettent d'apporter des réponses pertinentes à des problématiques diverses mettant en œuvre plusieurs choix possibles et de faciliter la prise de décision stratégique ou opérationnelle.

Nous allons développer dans ce paragraphe quelques méthodes couramment utilisées. Ces méthodes peuvent être classées en trois grands types : la résolution d'un problème par satisfaction de contraintes (CSP), l'aide à la décision multicritère et les méthodes basées sur l'intelligence artificielle.

2.3.5.1 Résolution d'un problème par satisfaction de contraintes

Un problème est décrit sous la forme d'un ensemble de contraintes sur des variables. Une contrainte exprime une relation entre différentes variables. Formellement, un problème de satisfaction de contraintes CSP (Constraint Satisfaction Problem) est défini par un triplet $\{X, \mathcal{D}, C\}$ où X sont des variables de décision, \mathcal{D} est un domaine de valeurs et C un ensemble fini de contraintes. Résoudre le problème consiste à trouver pour chaque variable $x_i \in X$ au moins une valeur $d_i \in \mathcal{D}$ respectant l'ensemble des contraintes C . Un CSP est cohérent s'il admet au moins une solution. S'il n'existe aucune solution, il faut relâcher certaines contraintes. Dans le cas contraire, s'il existe trop de solutions, il faut prendre la solution optimale, en ajoutant des contraintes supplémentaires ou en utilisant d'autres techniques d'optimisation (voir ci-après) (Moisan 2000; Moullec et al. 2012b; Trabelsi et al. 2013).

2.3.5.2 Aide à la décision multicritère

Comme de nombreux critères sont généralement nécessaires pour répondre à un objectif global, à critères multiples, des méthodes de prise de décision, dont les principes, avantages

et inconvénients sont largement détaillées dans certaines synthèses bibliographiques (Figueira, Greco et Ehrgott 2005; Bouyssou, 2006), permettent de trouver la meilleure solution ou au moins la meilleure solution dans un ensemble de solutions.

Les étapes de cette méthode sont (Lo 2013) :

1. Identifier l'ensemble des objectifs qui doivent être examinés
2. Modéliser les préférences du décideur
3. Evaluer les objectifs à l'aide de critères
4. Agréger les résultats.

Il existe trois grandes familles d'analyse multicritère selon la méthode d'agrégation : les méthodes d'agrégation complète, les méthodes d'agrégation de surclassement (agrégation partielle) et les méthodes d'agrégation locale.

- Pour la première famille, les méthodes consistent à agréger les scores de satisfaction entre eux puis à ranger les alternatives selon leur score, comme par exemple la méthode MAUT (Multi-Attribute Utility Theory), ou comparer les alternatives deux à deux puis agréger les résultats (analyse multicritère hiérarchique) avec par exemple la méthode AHP (Analytic Hierarchy Process) ou encore la méthode MACBETH (Measuring Attractiveness by a Categorical Based Evaluation TecHnique).

- Pour la seconde famille, les méthodes acceptent l'incompatibilité des alternatives et la non-transitivité des préférences (par exemple, il y a préférence de A par rapport à B, ainsi que de B par rapport à C, mais indifférence entre A et C). Les méthodes connues de cette famille sont les méthodes ELECTRE (ELimination Et Choix Traduisant la REalité) et la méthode PROMETHEE (Preference Ranking Organisation METHod for Enrichment Evaluations).

Pour la dernière famille, dans le cas où le nombre d'alternatives est très grand, la technique consiste à partir d'une solution de départ, à examiner si des solutions voisines ne sont pas meilleures (exemple : méthode STEM (STEp Method) réduit progressivement l'ensemble des actions en ajoutant des contraintes (matrice des gains) sur les valeurs des critères ou encore la méthode du point de mire).

Néanmoins, l'utilisation de ces méthodes de sélection pour l'évaluation quantitative des architectures, nécessitent que des critères/métriques spécifiques soient préalablement définis.

2.3.5.3 Méthodes basées sur l'intelligence artificielle

L'étape de choix de la structure d'un système à concevoir est toujours difficile à automatiser car il s'agit de manipuler des connaissances et d'intégrer de l'expérience voire de la créativité, ou globalement de manipuler de l'intelligence. Une solution est de mettre en place des bases de données regroupant diverses solutions et leurs qualités respectives ou encore d'utiliser l'intelligence artificielle qui est encore peu développée. L'outil principal de cette méthode est le système expert. Il vise à reproduire le raisonnement d'un expert. Il faut donc représenter la connaissance de l'expert par des modèles et des règles de conception. Plus il y a de règles, plus la méthode est efficace mais souvent les règles se contredisent et bloquent l'aide à la décision (Espanet 1999a).

Finalement, dans le cas des systèmes complexes impliquant différents mécanismes, différents modèles, différentes fonctions, différents domaines, pour réduire l'ensemble de toutes les combinaisons de choix technologiques possibles, il est alors possible d'utiliser des outils d'intelligence artificielle pour sélectionner le meilleur concept de design,

l'architecture, produit, etc. avec les valeurs métriques les plus élevés ou plus bas, à partir d'un espace de solution générée par les concepteurs de systèmes.

Or quelle que soit la méthode, elle fait généralement appel à l'établissement de critères, d'indicateurs ou de métriques (Moisan 2000).

3 Métriques d'architectures développées

Comme Lo (Lo 2013) le souligne dans sa thèse, « en conception multidisciplinaire, les spécialistes des différentes disciplines impliquées ont tendance à apporter des solutions en privilégiant leur point de vue métier. L'évaluation vue alors au niveau système, avec un objectif global et consensuel reste encore peu ou pas mise en œuvre ».

Par ailleurs, en accord avec Casner (Casner et al. 2013), qui propose de positionner le processus d'optimisation des systèmes mécatroniques en phase amont du cycle de conception pour réduire les coûts (itérations tardives, changement tardif de topologies/architectures très coûteux), nous souhaitons proposer à l'architecte système mécatronique des métriques d'architectures, lui permettant de faire les bons choix de conception dès les premières phases de conception. Ces métriques lui permettront en outre de justifier et tracer son choix, afin que les autres concepteurs aient ces informations au plus tôt.

Ainsi afin de répondre à ces deux besoins, nous proposons des métriques répondant à l'objectif d'intégration des systèmes mécatroniques. Pour cela, nous nous intéressons à l'intégration multi-domaine et l'intégration des niveaux d'abstraction, qui différencient les systèmes mécatroniques des autres systèmes. Si l'*intégration horizontale* (à un seul niveau d'abstraction) des différents domaines de la mécatronique est déjà une source de complexité croissante de ces systèmes, il devient également de plus en plus difficile d'avoir une vue d'ensemble du système, et donc nécessaire de décomposer le système à différents niveaux d'abstraction pour en réduire la complexité. Cette décomposition doit prendre en compte les différents niveaux successifs de sous-systèmes, qui devront être intégrés physiquement au sein du système global. Cette décomposition peut s'appuyer sur la modélisation hiérarchique décrite par la suite, pour faciliter la gestion de la complexité liée à l'*intégration verticale* de tous les sous-systèmes successifs du système.

Dans ce chapitre, nous proposons donc des indicateurs et des métriques pour la phase de préconception mécatronique permettant à l'architecte système de choisir, parmi plusieurs architectures candidates, celle qui répondra à un plus grand effort d'intégration.

Pour cela, nous avons construit deux métriques (Figure 2-14) :

- La première est une métrique d'intégration des niveaux d'abstraction (MMLI : Metric for Multi-Level Integration) d'un système mécatronique. S'appuyant sur les travaux de modélisation hiérarchique cités dans l'annexe 3, nous avons considéré 4 niveaux d'abstraction pour les systèmes mécatroniques : le système mécatronique complet, les sous-systèmes mécatroniques, les modules mécatroniques et pour finir les composants mono-domaines.
- La deuxième métrique est celle de l'intégration multi-domaine (MMDI : Metric for Multi-Domain Integration) basée sur l'intégration des domaines de la mécatronique, que nous avons définis au nombre de quatre : la mécanique, l'électronique,

l'informatique et l'automatique, mais qui peuvent être étendus à d'autres domaines (l'approche restant la même).

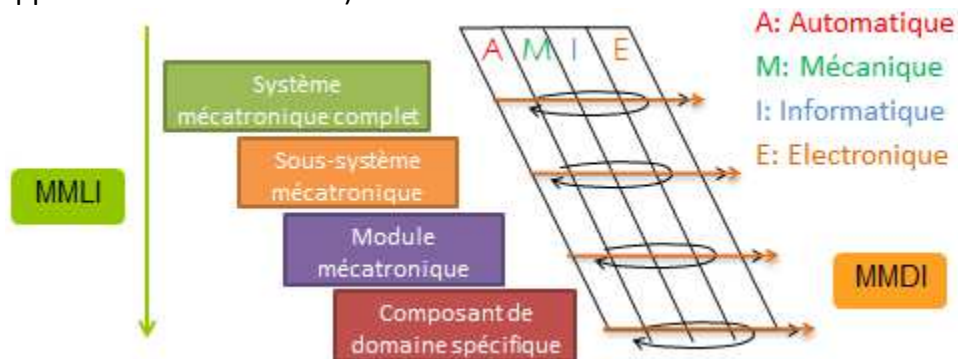


Figure 2-14 : Développement des métriques d'intégration multi-domaine et multi-niveau

3.1 Métrique d'intégration des niveaux d'abstraction

3.1.1 Enjeux

L'enjeu de la métrique sur l'intégration des niveaux d'abstraction est d'évaluer les risques/coûts ou au contraire les gains potentiels liés à l'intégration de nombreux composants ou sous-systèmes, en prenant en compte l'aspect progressif de cette intégration au niveau module, sous-système et système.

Cet enjeu peut être vu suivant trois principaux aspects : l'intégration physique de ces composants et ses conséquences, les aspects humains liés à la nécessité de collaborations et d'organisation rigoureuse que l'intégration multi-niveau rend déterminante, enfin les aspects techniques, stratégiques et organisationnels liés aux compromis d'arbitrage, dont la complexité est amplifiée par l'intégration multi-niveau.

L'intégration multi-niveau des niveaux d'abstraction d'un système se traduit matériellement par la nécessité d'intégration physique de nombreux composants au sein du système à concevoir. En effet, chaque composant va devoir être assemblé et interfacé à d'autres composants pour former un module, qui à son tour sera assemblé à d'autres modules, et ainsi de suite.

- Cela soulève d'abord un impact sur la *fabrication du système*. En effet, l'intégration multi-niveau va conditionner le planning, les techniques de fabrication (voir chapitre 3), les méthodes d'assemblage et de conditionnement/de sous-traitance du système. En effet, suivant la décomposition multi-niveau, l'entreprise pourra choisir de sous-traiter un module ou un sous-système complet, ou au contraire de n'acheter que les composants pour assurer lui-même l'assemblage... Quoiqu'il en soit, chaque élément fabriqué devra alors correctement s'interfacer avec les autres éléments du même niveau d'abstraction avec qui il a un lien logique. Cet aspect permet d'évaluer à la fois les efforts de sous-traitance, de logistique (transports des pièces), de conditionnement (emballage, quantités...), mais aussi les contraintes liées à l'ordonnancement de la fabrication des pièces (puisque pour monter d'un niveau, toutes les pièces du niveau inférieur doivent être disponibles...). De même, plus il y a de composants à assembler et plus le temps et le risque de difficulté d'assemblage s'accroissent. Les risques liés à l'interfaçage (tolérances géométriques, continuité physique (électrique, magnétique, thermique...)) vont être également proportionnel au nombre d'interfaces à réaliser, mais surtout à contrôler. Ces différents aspects

sont autant de source de coût potentiel que l'entreprise souhaite évaluer en amont pour prendre des décisions et faire des choix.

- Le deuxième aspect de l'intégration multi-niveau lié à l'intégration physique, est que l'un des principaux challenges des systèmes mécatroniques est d'intégrer un maximum de composants au sein d'un *volume de plus en plus petit*. En effet, intégrer et interfacier de nombreux composants alors que le volume alloué est de plus en plus réduit, posent le problème de la miniaturisation des composants et du coup de l'innovation mais aussi du coût, concernant ces nouveaux procédés/produits capables de réduire le volume d'encombrement des composants: surfaces fonctionnalisées, nanotechnologies, micro-mécatronique, matériaux intelligents...
- Un autre point important pour les industriels, qui est impacté par cette métrique d'intégration multi-niveau concerne directement le *cycle de vie du produit* : que ce soit pour l'accessibilité ou la démontabilité qui conditionnent directement la maintenabilité, la recyclabilité, mais également la modularité du produit. En effet, plus on a de composants à intégrer et plus la question d'accessibilité va se poser pour rajouter de nouveaux composants, au fur et à mesure de l'intégration multi-niveau. Une attention particulière devra être portée à l'ordre et à la technique d'assemblage, et son coût et ses risques associés, afin de permettre l'intégration complète du produit. Tout cela devra en plus tenir compte des besoins de maintenabilité et/ou de modularité envisagée par l'entreprise, suivant qu'elle préfère un produit jetable moins cher ou plus onéreux mais maintenable ou évolutif (modulaire). Ces aspects ont un impact direct sur l'architecture multi-niveau à choisir. Les enjeux liés à la modularité seront détaillées dans le chapitre 4. Enfin, une des difficultés actuelles des systèmes mécatroniques très intégrés est la recyclabilité des produits, car plus on « intègre » physiquement et plus il devient compliqué de séparer les différents matériaux lors du recyclage. Or cet enjeu risque de devenir un critère de plus en plus important à prendre en compte dans le choix des architectures dans les prochaines années.

En parallèle, la gestion des relations humaines et organisationnelles liée à cette intégration multi-niveau est également un point notable. En effet, la multitude et la diversité des composants à intégrer supposent de faire intervenir des équipes de spécialité différente, qui devront échanger, travailler et collaborer ensemble très régulièrement pour gérer efficacement les interfaces et autres difficultés liées à l'intégration physique des composants (voir § précédent). Il faudra donc veiller à l'organisation fréquente de jalons et autres réunions plus informelles entre les équipes « techniques /technologiques » pour anticiper et désamorcer tout risque lié à cette intégration. La métrique d'intégration multi-niveau permettra d'évaluer quels sont les niveaux d'intégration les plus risqués (forte intégration) et ainsi évaluer le coût correspondant. En effet, si les équipes sont divisées suivant leurs compétences sur tel ou tel type de composant, il faudra s'assurer que lorsqu'il faudra qu'elles interfacent leur composant avec une autre équipe, tout d'abord qu'elles se comprennent (vocabulaire employé, modèles utilisés), mais surtout qu'elles aient une vision commune (objectifs définis, paramètres communs, priorités...) du système. Cela suppose potentiellement de prévoir certains coûts de formations « transversales » préalables pour que chaque équipe ait les bases techniques suffisantes pour comprendre les arguments de l'autre équipe. Cet aspect est d'autant plus important pour les systèmes mécatroniques qui

font justement intervenir des composants multi-domaines. Un autre aspect humain et organisationnel est la nécessité d'accompagnement, de supervision de la gestion de ces interfaces qui sera d'autant plus grande que l'effort d'intégration sera élevé : il est beaucoup plus complexe de gérer l'intégration de 10 composants (planification : séquentielle et parallèle, approvisionnement, compromis...) que de 2 composants. Cette mission devra être prise en compte sur un poste budgétaire pour évaluer le coût de telle ou telle architecture.

Enfin, plus il y a de composants et de niveaux d'intégration, et plus le processus décisionnel va être complexe et les critères correspondants primordiaux pour parvenir à des compromis et limiter les risques. En effet, de manière générale, plus il y a de composants, plus il y aura d'interfaces à gérer et donc de décisions à prendre sur le « mode » d'interfaçage, autant de temps et de coût lié aux concertations, au temps de développement, de fabrication et d'assemblage, qu'il faudra évaluer pour choisir l'architecture optimale. A cela, il faut également prendre en compte d'autres objectifs techniques comme la fiabilité, la robustesse qui vont directement être impactés ou impacter le nombre de composants. Par exemple, tandis que la redondance de composants augmente la fiabilité, elle peut réduire la robustesse en augmentant le risque lié au grand nombre d'interfaces (mécaniques, électriques) et composants (connecteurs) associés. Par ailleurs, le nombre élevé de composants peut aller à l'encontre de l'optimisation volumique, de masse ou d'énergie, tant recherchée actuellement. Enfin, pour les systèmes mécatroniques ou autres systèmes dont la synergie/couplage des composants est effective, il est reconnu que l'optimisation globale du système ne correspond pas à la somme des solutions optimales locales de ses sous-systèmes. Autrement dit, le problème d'optimisation globale des performances sera d'autant plus complexe (et coûteux) que le nombre de composants à prendre en compte dans l'optimisation sera grand. Les compromis seront donc un élément important à prendre en compte en fonction de l'intégration multi-niveau de l'architecture choisie.

3.1.2 Métrique d'intégration multi-niveau du système

La métrique globale d'intégration multi-niveau du système est une valeur normalisée composée de deux indicateurs IML_s et IML_n . Cette métrique prend en compte l'aspect d'intégration multi-niveau global du système depuis le niveau d'abstraction le plus bas jusqu'au niveau le plus haut (IML_s), et les aspects d'intégration multi-niveau locaux entre deux niveaux successifs (IML_n). Pour définir notre métrique globale nous avons choisi d'agrèger ces indicateurs, en définissant une *fonction objectif* comme la somme de l'ensemble de ces indicateurs (Scaravetti 2004) pondérés par des facteurs λ_n positifs compris entre 0 et 1. La somme de ces coefficients λ_n doit rester égale à 1.

$$\begin{aligned} \text{Fonction objectif} \\ = \lambda_1 \text{Indicateur}_1 + \lambda_2 \text{Indicateur}_2 + \dots + \lambda_n \text{Indicateur}_n \end{aligned} \quad (2.3)$$

Les coefficients λ_n sont préalablement univoquement déterminés par les priorités du concepteur/architectes/décideurs ou leur retour d'expériences et permettent ainsi de donner un poids plus important à certains indicateurs dans la fonction objectif.

Pour pondérer ces indicateurs, nous pouvons alors nous appuyer sur les techniques d'aide à la décision décrites précédemment (§2.3.5).

$$MMLI = \sum_{n=1}^{nl-1} \lambda_n IML_n + \lambda_s IML_s \quad (2.4)$$

Avec nl : nombre de niveaux considérés. Dans notre cas, nous avons identifié 4 niveaux (le système complet, les sous-systèmes, les modules, les composants),
 n : le niveau supérieur considéré.

Les coefficients λ_x sont préalablement déterminés selon les priorités de conception. La somme de ces coefficients doit être égale à 1. Le concepteur peut privilégier le coefficient λ_s , s'il souhaite avoir une vision globale de l'intégration multi-niveau sur un système, ou il peut privilégier un des coefficients λ_n pour souligner un effort d'intégration plus localisé entre deux niveaux d'abstraction donnés.

3.1.3 Indicateur d'intégration des niveaux d'abstraction pour le système : IML_s

3.1.3.1 Définition

Le premier indicateur IML_s est un indicateur global d'intégration multi-niveau pour le système. Il évalue si l'intégration globale entre le premier et le dernier niveau d'abstraction.

3.1.3.2 Expression

L'expression de cet indicateur est la suivante :

$$IML_s = \prod_{n=1}^{nl-1} \left(1 - \frac{n_e}{n_{e_{n-1}}} \right) \quad (2.5)$$

Avec :

nl : nombre de niveaux considérés. Dans notre cas, nous avons identifié 4 niveaux (le système complet, les sous-systèmes, les modules, les composants).

n_e : nombre total d'éléments dans l'architecture au niveau d'abstraction n .

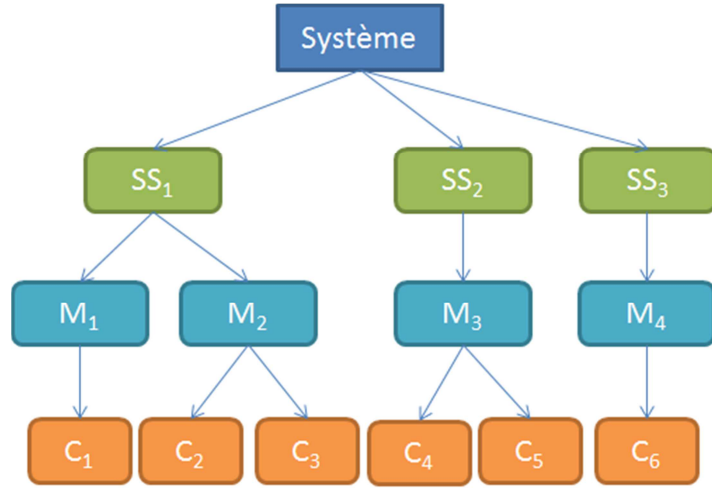
$n_{e_{n-1}}$: nombre d'éléments du niveau $n-1$.

3.1.3.3 Bornes

Comme précédemment, la borne inférieure est égale à 0 et la borne supérieure tend vers 1. La valeur vaut 0 lorsqu'il n'y aucune intégration entre les différents niveaux. La valeur approche 1 lorsque quand il y a un important effort d'intégration dans tous les niveaux.

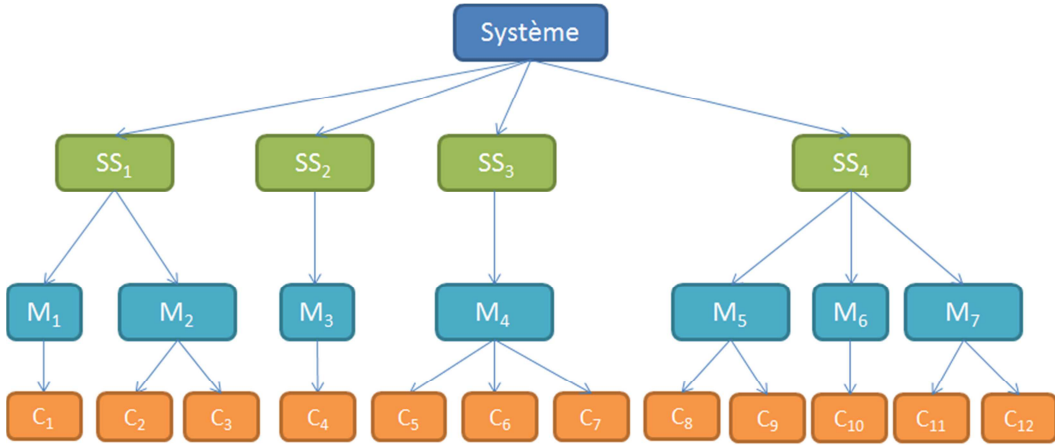
3.1.3.4 Exemple

Pour illustrer cet indicateur, nous prenons un exemple théorique simple. Les différents niveaux sont représentés par « Système » pour le niveau système, par « SS_x » pour les composants du niveau sous-système, par « M_x » pour les éléments du niveau module et par « C_x » pour les constituants du niveau composant. Le premier exemple (Figure 2-15) comporte 6 composants, 4 modules et 3 sous-systèmes pour composer le système (A). Le deuxième système (Figure 2-16), le système (B), a deux fois plus de composants, il est aussi composé de 7 modules et de 4 sous-systèmes.



(A)

Figure 2-15 : Premier exemple d'architecture candidate



(B)

Figure 2-16 : Deuxième exemple d'architecture candidate

Les résultats pour l'indicateur global d'intégration multi-niveau sont les suivants :

- Pour le système (A)

$$IML_S = \left(1 - \frac{1}{3}\right) \times \left(1 - \frac{3}{4}\right) \times \left(1 - \frac{4}{6}\right) = \frac{1}{18} \cong 0,055$$

- Pour le système (B)

$$IML_S = \left(1 - \frac{1}{4}\right) \times \left(1 - \frac{4}{7}\right) \times \left(1 - \frac{7}{12}\right) = \frac{15}{112} \cong 0,134$$

3.1.4 Indicateur d'intégration des niveaux d'abstraction par niveaux successifs : $IMLn$

3.1.4.1 Définition

Le second indicateur est un indicateur d'intégration multi-niveau entre deux niveaux consécutifs. Il permet de mesurer le nombre d'éléments d'un niveau n intégré dans le niveau supérieur n . Il existe donc $n - 1$ indicateurs. Dans notre cas, conformément à la

modélisation hiérarchique appliquée à la mécatronique proposée par (Hehenberger 2012), nous avons défini 4 niveaux d'abstraction, qui nous donnent donc 3 indicateurs IML_n . L'indicateur IML_1 pour l'intégration multi-niveau entre le niveau des sous-systèmes et le niveau du système complet. L'indicateur IML_2 pour l'intégration multi-niveau entre le niveau des modules et le niveau des sous-systèmes. Et le dernier, l'indicateur IML_3 , pour l'intégration multi-niveau entre le niveau des composants et le niveau des modules.

3.1.4.2 Expression

L'expression de cet indicateur (Figure 2-17) est donc la suivante :

$$IML_n = \frac{\sum_{k=1}^{n_e} \left(1 - \frac{1}{n_{al_{ek}}}\right)}{n_e} \quad (2.6)$$

Avec n : le niveau d'abstraction considéré

n_e : le nombre total d'éléments dans l'architecture au niveau d'abstraction n .

$n_{al_{ek}}$: le nombre d'éléments e_k d'une architecture du niveau $n-1$, alloués à l'élément k de l'architecture de niveau n .

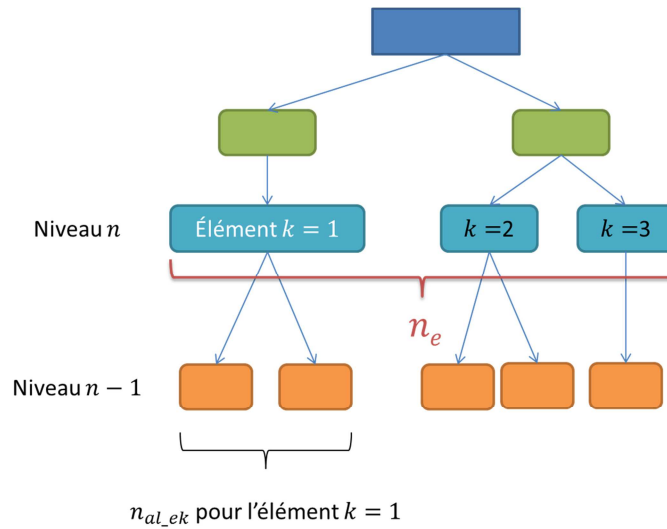


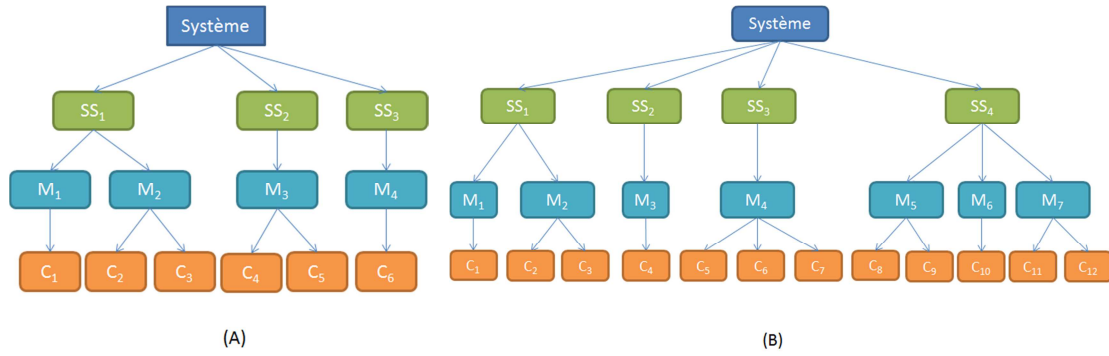
Figure 2-17 : Schématisation des termes de la formule (2.6)

3.1.4.3 Bornes

La valeur de cet indicateur est toujours pour la borne inférieure : 0 et la borne supérieure tend vers 1. La valeur 0 est atteinte lorsqu'il n'y a aucune intégration entre le niveau $n-1$ et le niveau n . C'est-à-dire qu'un élément de niveau $n-1$ correspond à un élément de niveau n . La valeur tend vers 1 quand de nombreux éléments d'un niveau $n-1$ sont intégrés dans un seul élément de niveau n .

3.1.4.4 Exemple

En reprenant l'exemple du paragraphe Chapitre 2 §3.1.3.4, nous obtenons les résultats suivants :



Pour le système (A), nous obtenons :

Au niveau Système complet / Sous-systèmes: $IML_1 = \frac{1 - \frac{1}{3}}{1} = \frac{2}{3} \cong 0,667$

Au niveau Sous-systèmes / Modules: $IML_2 = \frac{1 - \frac{1}{2} + 1 - \frac{1}{1} + 1 - \frac{1}{1}}{3} = \frac{1}{6} \cong 0,167$

Au niveau Modules / Composants: $IML_3 = \frac{1 - \frac{1}{1} + 1 - \frac{1}{2} + 1 - \frac{1}{2} + 1 - \frac{1}{1}}{4} = \frac{1}{4} = 0,25$

Pour le système (B), nous obtenons les résultats suivants :

Au niveau Système complet / Sous-systèmes: $IML_1 = \frac{1 - \frac{1}{4}}{1} = \frac{3}{4} = 0,75$

Au niveau Sous-systèmes / Modules: $IML_2 = \frac{(1 - \frac{1}{6}) + (1 - \frac{1}{1}) + (1 - \frac{1}{1}) + (1 - \frac{1}{3})}{4} = \frac{7}{24} \cong 0,29$

Au niveau Modules / Composants :

$$IML_3 = \frac{(1 - \frac{1}{1}) + (1 - \frac{1}{2}) + (1 - \frac{1}{1}) + (1 - \frac{1}{3}) + (1 - \frac{1}{2}) + (1 - \frac{1}{1}) + (1 - \frac{1}{2})}{7} = \frac{13}{42} \cong 0,309$$

3.1.5 Résultats, analyse et discussions

En reprenant la formule (2.4) , la métrique vaut alors, avec des coefficients équilibrés entre les deux indicateurs ($\lambda_n = 1/6$ pour $n=1$ à 3 et $\lambda_s = 0,5$):

$$MMLI = \sum_{n=1}^{nl-1} (\lambda_n IML_n) + \lambda_s IML_s$$

- Pour le système (A) :

$$MMLI_{(A)} = \frac{1}{6} \times \frac{2}{3} + \frac{1}{6} \times \frac{1}{6} + \frac{1}{6} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{18} = \frac{15}{72} \cong 0,21$$

- Pour le système (B) :

$$MMLI_{(B)} = \frac{1}{6} \times \frac{3}{4} + \frac{1}{6} \times \frac{7}{24} + \frac{1}{6} \times \frac{13}{42} + \frac{1}{2} \times \frac{15}{112} = \frac{589}{2016} \cong 0,292$$

Le graphe ci-dessous représente les résultats des deux systèmes en fonction des indicateurs et les résultats des métriques :

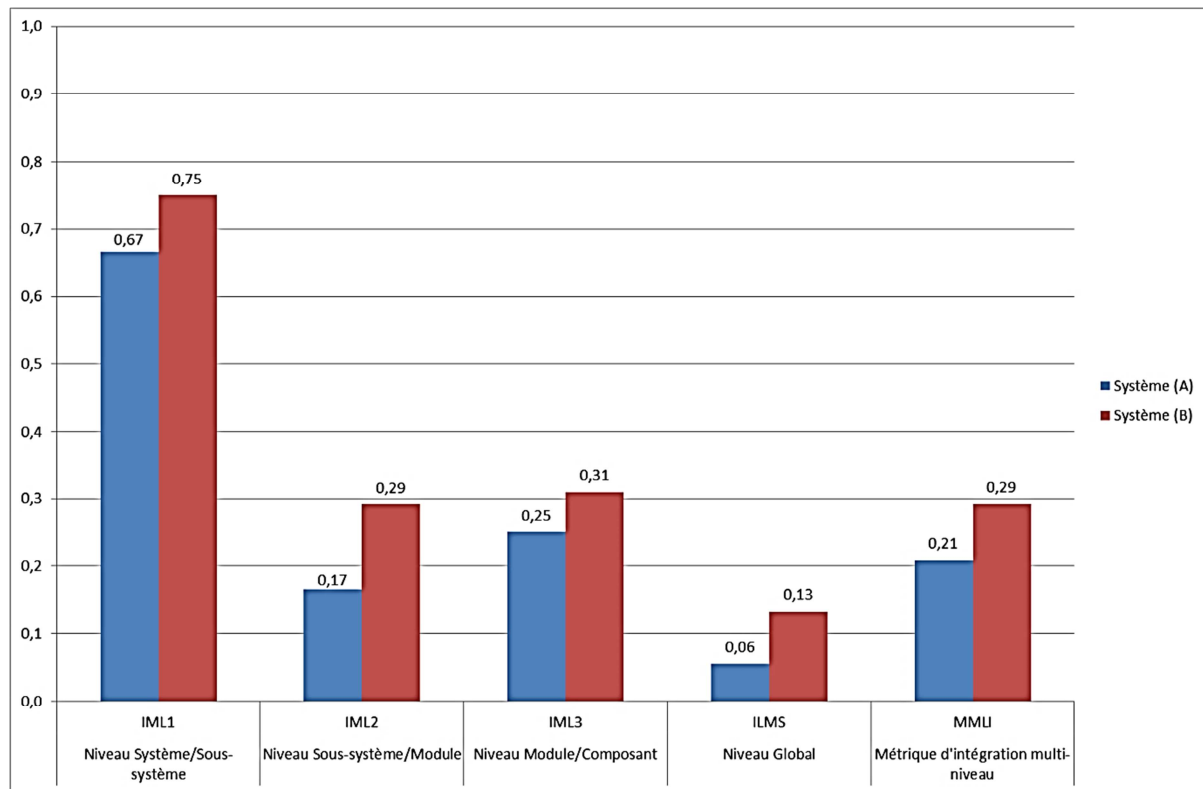


Figure 2-18 : Résultats des deux architectures candidates en fonction des indicateurs

De manière générale, le système (B) est plus intégré que le système (A). Que ce soit en ce qui concerne les niveaux d'abstraction, où l'indicateur d'intégration IML_x indique l'effort d'intégration entre chaque niveau successif, que pour l'indicateur global IML_s , plutôt relatif aux nombres globaux de composants intégrés sur l'ensemble des niveaux. On note en effet qu'au niveau des intégrations successives, les indicateurs IML affichent un écart moyen de 25% d'intégration en moins pour le système (A), et au niveau global un écart des $ILMS$ de 60% de moins pour le système (A). Effectivement, l'effort d'intégration local est quasi similaire pour les deux systèmes, excepté sur le niveau 2 grâce au sous-système SS4 du système (B). Concernant l'effort d'intégration global du système, le plus fort écart provient du fait que le système(B) possède deux fois plus de composants à intégrer que le système (A).

Ces indicateurs sont importants à prendre en compte pour évaluer les risques/coûts liés à l'intégration physique de tous ces composants tout au long du projet, que ce soit dans la gestion des interfaces, que dans le placement de ces composants pour la fabrication, mais également pour la maintenabilité et le recyclage du produit, les couplages multi-physiques pouvant apparaître à cause de la proximité de composants multi-domaines, sans oublier tous les aspects collaboratifs et de compromis inhérents à la multitude et à la diversité de ces composants... Par exemple, pour la fiabilité/sécurité d'un système un compromis sera à trouver entre fiabilité et faible intégration. En effet, les redondances (nombre de composants élevé) améliorent la fiabilité mais augmente la complexité liée à une intégration multi-niveau plus importante.

Dans l'exemple, nous avons appliqué la métrique ainsi que les deux indicateurs sur une décomposition hiérarchique (Système/Sous-système/Module/Composant) du système.

Cependant, cette métrique peut être également appliquée à d'autres modélisations hiérarchiques comme par exemple celle des niveaux d'architectures fonctionnelle/logique/organique/physique, sous réserve qu'elles soient sous forme d'arbre (chaque élément d'un niveau doit être décomposé en un ou plusieurs éléments au niveau inférieur, le nombre d'éléments au niveau n doit donc être inférieur au nombre d'éléments du niveau inférieur).

Cette métrique peut également être étendue à des décompositions ayant un nombre différents de niveaux d'abstractions, il suffit de modifier la valeur de nl et de calculer les indicateurs « locaux » pour chaque niveau.

De plus, dans le cas où un élément ne serait pas décomposé en un sous-élément, l'indicateur d'intégration multi-niveau au niveau du système IML₅ serait alors nul, cela signifierait qu'aucun effort de décomposition/intégration ne serait fait. Enfin cette métrique peut être couplée avec d'autres métriques comme les métriques d'intégration multi-domaine ou d'innovation qui seront détaillées dans les paragraphes suivants. En effet, pour chaque niveau d'abstraction, il est alors possible de calculer une métrique multi-domaine et/ou d'innovation qui permet d'avoir une vision « locale » de l'intégration. Ainsi l'architecte système aura une vision des couplages des différents domaines ou d'effort d'innovation tout au long des étapes d'intégration des éléments.

3.2 Métrique d'intégration multi-domaine

3.2.1 Enjeux

Cette métrique a été développée pour évaluer l'intégration des domaines de la mécanique dans le système en prenant en compte les différents niveaux d'abstraction. L'enjeu de l'intégration multi-domaine est de mesurer au travers de l'aspect multi-domaine d'un système mécanique, son degré d'innovation et le gain en volume, mais en parallèle le risque/les coûts liés à cette *complexité* (Chapitre 1 §4.2.3). En effet, l'intégration multi-domaine augmente la complexité du système à concevoir car elle fait intervenir la complexité intrinsèque des différentes disciplines impliquées à laquelle se rajoute celle des interactions et couplages entre ces disciplines.

Considérons tout d'abord les avantages d'une forte intégration multi-domaine, elle est à l'origine même du développement de la discipline mécanique depuis presque 50 ans et ainsi de toute l'innovation des produits qui en résulte. Aujourd'hui, rares sont les produits de notre quotidien (électro-ménager, loisirs, de notre environnement professionnel..) qui ne font pas appel à au moins 3 domaines de la mécanique. Tout notre environnement est rempli de capteurs, d'automatismes, d'« intelligence » sur des systèmes qui autrefois étaient purement mécaniques... Ainsi après avoir d'abord porté l'innovation sur la juxtaposition/communication entre ces disciplines, le challenge actuel est plus à l'optimisation de ces systèmes pour intégrer toujours plus de fonctions, mais dans un volume toujours réduit... (Figure 2-19).

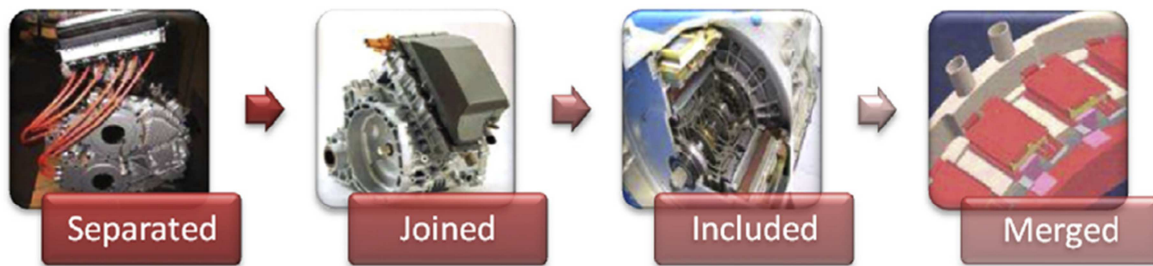


Figure 2-19 : Les différents niveaux d'intégration dans les systèmes mécatronique (Projet O2M : Outils de Modélisation Mécatronique)³²

Au-delà de nouveaux produits innovants (solowheel³³, tablettes, google glass..) qui ont développé de nouveaux besoins et habitudes de notre quotidien, et donc de nouveaux marchés potentiels, l'intégration multi-domaine des systèmes mécatroniques a aussi permis d'ajouter de nombreuses fonctionnalités à des objets existants (roulement instrumenté, véhicule autonome, smartphone, montre google). Ainsi l'intégration multi-domaine, au travers de la dématérialisation de composants électroniques par des logiciels embarqués, par l'intégration de composants externes appartenant à d'autres disciplines à l'intérieur d'un même système, et de l'automatisation et du contrôle de certaines fonctions mécaniques par l'ajout de capteurs et d'actionneurs, est un fort vecteur d'innovation actuelle. Ainsi, les produits deviennent toujours plus intelligents et grâce à l'augmentation croissante de la puissance informatique, les fonctions sont de plus en plus numérisées et par conséquent en constante augmentation.

Enfin, une autre conséquence intéressante de cette intégration multi-domaine est la réduction du volume des systèmes, qui permet souvent des économies de masse et de coût (manutention, transport, stockage/étalage,...). Par ailleurs cette intégration a fait naître de nouvelles technologies comme la technologie LDS (Laser Direct Structuring) ou encore les MID (Model Interconnect Device).

En parallèle, le coût sera impacté par la complexité liée au couplage entre les disciplines (couplage multi-domaine), accentuée par la structure hiérarchique d'intégration des différents éléments (multi-domaines) au sein d'un ensemble (couplage multi-niveau). Nous avons vu précédemment que la mécatronique était la synergie d'au moins quatre domaines qui sont la mécanique, l'électronique, l'informatique et l'automatisme, or les aspects de la complexité multi-domaine qui vont être détaillés par la suite vont être accrus par le nombre de domaines et la complexité de chaque domaine.

Nous avons décliné cette complexité suivant deux aspects : l'aspect technique et l'aspect humain.

Concernant l'aspect purement technique, la *gestion des interfaces* entre composants de domaines différents, la *différence d'objectifs/performances* et la *réduction du volume* vont être les principales contraintes impactant cette complexité.

- La gestion des interfaces va se traduire par le besoin de partager les informations (modèles, données, paramètres) pendant la conception et de communiquer (gestion des « connections ») pendant la phase de fonctionnement. Ainsi la nécessité de gérer

³² Projet O2M (Outils de Modélisation Mécatronique) labellisé par Mov'eo et System@tic, financé par le FUI (Fond Unique Interministériel)

³³ <http://www.solowheel.fr/>

l'interopérabilité des outils, des langages (à défaut de trouver un langage commun, auquel il faudra former les personnes quoiqu'il en soit), pour partager les modèles et à minima partager les paramètres communs), mais aussi de trouver des points de convergence entre les différentes méthodes de conception, pour assurer un processus de conception continu et cohérent, rend le processus de conception multi-domaine plus coûteux. En effet, il est difficile de trouver un outil permettant de gérer les maquettes 3D de mécanique et les schémas 2D électroniques pour faire une conception ECAD/MCAD complètement 3D, qui garde les fonctionnalités des outils métiers d'origine. Par ailleurs, l'implémentation de composants mono-domaines au sein d'un même sous-ensemble implique la prise en compte des connecteurs en phase amont, pour éviter que ce « détail » n'entrave la validité de la conception trop tardivement avec des surcoûts exorbitants. Cela signifie que les « câbleurs » et les personnes qui s'occupent des connecteurs doivent intégrer les équipes de conception préliminaire au plus tôt pour faire émerger leurs contraintes dans le cahier des charges partagé aux autres concepteurs. Cette problématique des connecteurs est d'ailleurs très étroitement liée au dernier point sur la réduction de volume.

- La différence d'objectifs entre les électroniciens, les mécaniciens, les informaticiens et les automaticiens est évidente : tandis que les uns vont s'occuper plus de la fiabilité, redondance du contrôle, d'autres s'attacheront plus aux problématiques d'encombrement et de cinématique, alors que d'autres s'attacheront plus à l'aptitude/rapidité à la mise à jour du code (« update »/ « release »).
- Enfin, la réduction du volume implique que les composants des différents domaines sont de plus en plus proches, voire physiquement imbriqués (Chapitre 3). Cette proximité va créer des couplages multi-physiques qui requièrent une attention et des études spécifiques supplémentaires (donc coûteuses) pendant le processus de conception. Ces couplages multi-physiques font intervenir l'ensemble des composants des différents métiers : par exemple, la carte électronique qui en chauffant, dilate le support mécanique ou endommage d'autres composants trop proches, de la même manière pour le rayonnement électromagnétique qui par la proximité des composants risque de compliquer la gestion de la CEM du système... Ainsi, que ce soit dans le processus de conception, qui va devoir intégrer des simulations multi-physiques répétées à chaque changement (de dimension, de matériaux, de position), très coûteuses (en temps, car généralement basées sur la modélisation en éléments finis), ou dans l'achat de logiciels multi-physiques performants, ce couplage multi-physique dû à l'intégration physique de composants multi-domaines a directement un impact sur le coût lié à l'aspect multi-domaine des systèmes mécatroniques. Au-delà de ces couplages, la réduction du volume des systèmes va également pousser à la miniaturisation des composants. Nous pouvons également citer les micro- (MEMS, micro-usine..) et nanotechnologies qui sont un fort vecteur d'innovation et de développement, mais qui impliquent de nouvelles technologies de fabrication, et donc d'investir dans de nouvelles machines de production/fabrication, de nouveaux processus. La miniaturisation des connecteurs et des câbles reste encore un sujet de recherche actuel très en vogue, qui n'a pas

encore atteint le même niveau de miniaturisation que les composants et restent donc un frein (coûteux) au développement de produits mécatroniques.

Concernant l'aspect humain, on retrouve la complexité classique liée au besoin de faire travailler des équipes différentes : avec des différences d'outils, méthodes, langages/vocabulaires, de compétences, d'objectifs, de sites géographiques, qui implique des changements dans leur manière de travailler. En effet, que ce soit dans le processus de conception qui nécessite des jalons de synchronisation plus nombreux et plus critiques, dans le processus d'arbitrage des solutions techniques, qui va mettre en lumière des objectifs contradictoires suivant la discipline, que dans la compréhension des données/modèles des autres « métiers », la nécessité de partager des informations communes et de travailler ensemble avec des personnes qui « ne parlent pas le même langage » est lourd de conséquence au niveau des coûts. Cette ingénierie collaborative permet d'intégrer tous les métiers techniques au projet mais également tous les sites géographiques ainsi que tous les intervenants (sous-traitants, fournisseurs,...). Lorsqu'un projet est partagé, les difficultés supplémentaires (et les coûts correspondants) peuvent apparaître lors de l'assemblage, de la fabrication, des interfaces, des différentes normes de chaque pays,... De son côté, le service des Ressources humaines est également directement impacté par cet aspect : que ce soit dans le besoin de recruter des personnes au profil « mécatronique » plus transversal pour assurer le lien et la compréhension entre les différents services, que lors de la formation d'agents, qui ont fait de la conception mono-domaine pendant de longues années, et qui doivent désormais intégrer d'autres contraintes techniques, un nouveau vocabulaire « multidisciplinaire », des méthodes de conception plus collaboratives, et donc changer leur façon de travailler pour évoluer vers un nouveau métier plus « mécatronique ». De la même façon, dans la gestion de projet, un projet pluridisciplinaire va demander peut-être un temps de développement plus long et/ou avec plus d'effort de supervision pour gérer justement ces « interfaces » entre personnes de domaine différents.

3.2.2 Métrique d'intégration multi-domaine du système

La métrique d'intégration multi-domaine est composée d'un indicateur pour chaque niveau hiérarchique. Le premier est l'indicateur d'intégration multi-domaine au niveau du système global et les suivants représentent celui de l'intégration multi-domaine au niveau local pour un seul niveau d'abstraction donné. En effet, il peut être intéressant pour l'architecte système de connaître le niveau d'intégration multi-domaine d'un module mécatronique, d'un sous-système mécatronique ou des composants, car ces mesures auront des conséquences différentes suivant le point de vue considéré (planning, ressources techniques et logicielles, compétences, formations, organisation, jalons de conception), comme nous l'avons détaillé précédemment.

Avec ces quatre indicateurs décrits ci-après, nous construirons une métrique d'intégration multi-domaine du système, qui est une valeur normalisée, prenant en compte les aspects multi-domaines locaux au niveau de chaque élément de chaque niveau et global pour un système donné.

Pour pondérer ces quatre indicateurs, nous pouvons alors nous appuyer sur les techniques d'aide à la décision décrites précédemment (§2.3.5). Pour appliquer cette métrique, nous avons utilisé une simple somme pondérée normalisée.

$$MMDI = \sum_{j=1}^{nd} \lambda_j IMD_j \quad (2.7)$$

Les coefficients λ_i sont déterminés par le concepteur selon ses priorités de conception. La somme de ces coefficients est égale à 1. Le concepteur peut privilégier le coefficient λ_1 , s'il veut connaître si un système a globalement une architecture multi-domaine ou il peut privilégier les coefficients λ_i ($1 \leq i \leq 3$) s'il souhaite mettre l'accent sur l'intégration multi-domaine plus localement dans chaque élément (composant, module, sous-système) d'un niveau d'abstraction donné.

3.2.3 Indicateur d'intégration multi-domaine sur un niveau d'abstraction n : IMD_n

3.2.3.1 Définition

L'indicateur mesure l'intégration des domaines de référence de la mécanique (Automatique, Mécanique, Informatique et Electronique) adressés par les éléments d'un niveau n . Dans notre cas, où nous avons distingué 4 niveaux, l'indicateur permet donc de mesurer l'intégration des domaines de la mécanique dans les éléments appartenant aux 4 niveaux d'abstraction mécaniques suivants: le système, les sous-systèmes, les modules et les composants.

Notons toutefois que le premier indicateur d'intégration multi-domaine du système (IMD_1) indique si les domaines de référence de la mécanique sont globalement présents dans le système, il correspond donc à une vision globale de l'indicateur, puisque la décomposition hiérarchique du système impose que la somme des domaines des éléments d'un niveau inférieur se retrouve dans l'élément « père » du niveau supérieur d'abstraction.

Par ailleurs, le calcul de cet indicateur au niveau 4, ne donnera des indications que sur la présence ou non de composants mono-domaines hors des domaines de référence de la mécanique prédéfinis précédemment.

3.2.3.2 Expression

L'expression de cet indicateur est donc la suivante :

$$IMD_j = \frac{\frac{1}{nd} \sum_{i=1}^{n_e} card(D_i)}{n_e} \quad (2.8)$$

Avec n : le niveau d'abstraction considéré

nd : le nombre total de domaines de référence. Dans notre cas nd vaut 4 puisque nous considérons 4 domaines de la mécanique: la mécanique, l'électronique, l'informatique et l'automatique.

D_i : l'ensemble des domaines de référence de la mécanique adressés dans l'élément i .

n_e : le nombre total d'éléments dans l'architecture au niveau d'abstraction n .

3.2.3.3 Bornes

Conformément aux spécifications que nous nous sommes fixées pour l'élaboration des métriques (Chapitre 1 §3.2.1) et notamment sur l'aspect normalisation, les bornes de cet indicateur sont 0 et 1, mais elles dépendent du niveau d'abstraction n considéré.

Pour le niveau système (IMD_1) comme il n'y a qu'un seul élément, la valeur 1 est obtenue lorsque tous les domaines de référence sont présents au moins une fois dans n'importe quel

élément quel que soit le niveau considéré. La valeur 0 est obtenue lorsqu'aucun domaine de la mécanique n'est présent dans les éléments quel que soit le niveau considéré.

Pour le niveau composant (IMD_4), comme les composants sont mono-domaines, il ne peut y avoir qu'un seul domaine par élément. S'il n'y a aucun autre domaine que les quatre de la mécanique, l'équation s'écrit $IMD_4 = \frac{\frac{1}{4} \sum_{i=1}^{n_e} 1}{n_e} = \frac{\frac{n_e}{4}}{n_e} = \frac{1}{4}$, quel que soit le nombre d'éléments, ainsi la valeur de la borne maximale de cet indicateur IMD_4 est égale à $\frac{1}{4}$.

3.2.3.4 Exemple

Pour illustrer cet indicateur, nous reprenons l'exemple théorique simple du paragraphe 3.1.3.4. Cette fois-ci, les deux architectures candidates ont été enrichies par les domaines techniques de chaque élément constituant le système. Les domaines de référence sont représentés par les lettres M (Mécanique), E (Electronique), I (Informatique) et A (Automatique) dans chaque élément. Dans le premier exemple, le système comporte tous les domaines de la mécanique. Dans le second, il n'y a que la mécanique et l'électronique qui sont représentés, plus deux autres domaines qui sont indépendants (qui n'ont aucun couplage avec les autres domaines de la mécanique) ou simples à traiter, de sorte qu'ils n'impactent pas ou peu les problématiques liées à l'intégration multi-domaine. Ces deux domaines sont les télécoms et l'hydraulique, et seront identifiés comme « Hors-domaine », avec la lettre H. Notons qu'il est parfois difficile de « trancher » sur le domaine d'un composant (le moteur en est un bon exemple). Dans l'industrie, il faudra prendre soin de faire cette évaluation en concertation avec les ingénieurs experts dans les domaines, et de tracer pour chaque « type » de composant, le domaine choisi, afin de pouvoir comparer différents produits, sur des sites géographiques différents ou entre plusieurs générations de produits.

Dans la pratique, le dernier niveau de décomposition à prendre en compte pour cette métrique est le niveau à partir duquel tous les éléments deviennent mono-domaines. Ainsi, par exemple, un frein comprenant deux éléments « mécaniques » (un disque de frein et un étrier) ne sera pas forcément « décomposé » pour évaluer l'intégration multi-domaine de l'architecture, contrairement à la métrique pour évaluer l'intégration multi-niveau précédente.

La figure ci-dessous reprend les exemples du paragraphe précédent enrichis avec leur domaine technique correspondant :

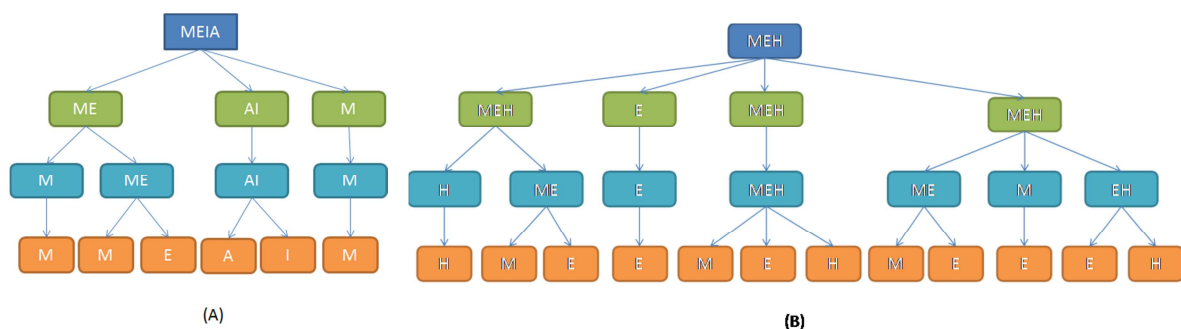


Figure 2-20 : Exemples théoriques avec enrichissement des domaines des éléments

Si nous reprenons l'exemple précédent, nous obtenons les résultats suivants :

Pour le premier système (système A), nous obtenons les résultats suivants :

- Au niveau système/global : $IMD_1 = \frac{\frac{1}{4}(4(MEIA))}{1} = 1$
- Au niveau des sous-systèmes : $IMD_2 = \frac{\frac{1}{4}(2(ME)+2(AI)+1(M))}{3} = \frac{5}{12} \cong 0,417$
- Au niveau des modules : $IMD_3 = \frac{\frac{1}{4}(1(M)+2(ME)+2(AI)+1(M))}{4} = \frac{3}{8} = 0,375$
- Au niveau des composants : $IMD_4 = \frac{\frac{1}{4}(1(M)+1(M)+1(E)+1(A)+1(I)+1(M))}{6} = \frac{1}{4} = 0,25$

Pour ce deuxième système (système B), nous obtenons les résultats suivants :

- Au niveau système/global : $IMD_1 = \frac{\frac{1}{4}(2(ME))}{2} = \frac{1}{2} = 0,5$
 - Au niveau des sous-systèmes : $IMD_2 = \frac{\frac{1}{4}(2(ME)+1(E)+2(ME)+2(ME))}{4} = \frac{7}{16} \cong 0,437$
 - Au niveau des modules : $IMD_3 = \frac{\frac{1}{4}(2(ME)+1(E)+2(ME)+2(ME)+1(M)+1(E))}{7} = \frac{9}{28} \cong 0,321$
 - Au niveau des composants :
- $$IMD_4 = \frac{\frac{1}{4}(1(M) + 1(E) + 1(E) + 1(M) + 1(E) + 1(M) + 1(E) + 1(M) + 1(E))}{12} = \frac{3}{16} = 0,187$$

3.2.4 Résultats, analyse et discussions

En reprenant la formule (2.7) , la métrique vaut alors, avec des coefficients équilibrés entre l'indicateur global IMD_1 et les indicateurs locaux IMD_2 , IMD_3 , IMD_4 ($\lambda_1 = 0,5$ et pour $n=2$ à 4 et $\lambda_n = 1/6$):

$$MMDI = \sum_{n=1}^{nd} \lambda_n IMD_n$$

Dans l'exemple précédent:

- Pour le système (A) :
 $MMDI_{(A)} = \frac{1}{2} \times 1 + \frac{1}{6} \times \frac{5}{12} + \frac{1}{6} \times \frac{3}{8} + \frac{1}{6} \times \frac{1}{4} = \frac{2}{3} = 0,68$
- Pour le système (B) :
 $MMDI_{(B)} = \frac{1}{2} \times \frac{1}{2} + \frac{1}{6} \times \frac{7}{16} + \frac{1}{6} \times \frac{9}{28} + \frac{1}{6} \times \frac{3}{16} = \frac{2}{5} \cong 0,40$

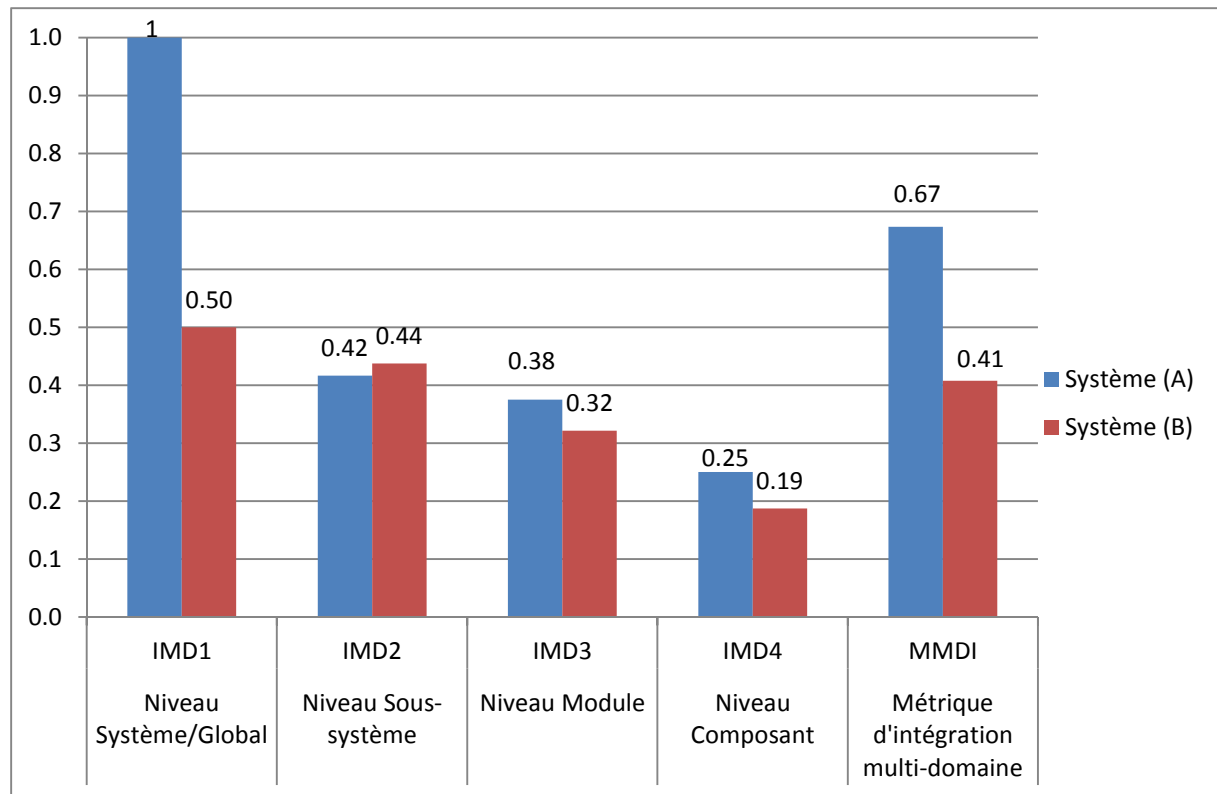


Figure 2-21 : Synthèse des résultats pour l'intégration multi-domaine des deux architectures candidates

Le système (A) atteint la valeur maximale de $\frac{1}{4}$ pour le niveau composant puisque tous les domaines de la mécanique et qu'aucun autre domaine n'est présent. L'indicateur d'intégration multi-domaine au niveau système vaut alors la valeur maximale de 1.

Pour le système (B), l'indicateur d'intégration multi-domaine au niveau IMD_1 montre que seule la moitié des domaines de la mécanique sont présents (les domaines de la mécanique et de l'électronique) et l'indicateur au niveau composant révèle qu'il y a, en plus, d'autres domaines (les télécoms et l'hydraulique) que ceux de la mécanique. En effet, les deux domaines de la mécanique représentent 75% des composants du système. L'indicateur au niveau sous-système affiche une plus forte intégration des domaines de la mécanique pour le système (B) que pour le système (A). Cela vient du fait qu'il y a plus de sous-systèmes intégrant des domaines de la mécanique (quelle que soit la variété des domaines) qui sont intégrés dans le système. Effectivement, pour le système (A), il y a 5 occurrences de domaine pour 3 sous-systèmes et pour le système (B) il y a 7 occurrences de domaine pour 4 sous-systèmes.

Cette métrique peut être utilisée durant deux phases du cycle de vie des architectures : pour les architectures organiques, lorsqu'on choisit les différentes technologies, et physiques, lorsque les composants sont définitivement spécifiés, mais d'une manière générale les domaines de la technologie seront les mêmes que les composants physiques, sauf qu'au niveau de décomposition hiérarchique du système, un composant physique (COTS par exemple) pourra intégrer plusieurs organes (technologies), et donc rajoutera un niveau de décomposition supplémentaire pour atteindre le niveau « mono-composant ».

Cette métrique peut être couplée avec la métrique d'innovation permettant de connaître l'effort à fournir dans un domaine donné pour mettre en œuvre un projet mécanique en termes de compétences, de formation et de quantité de travail.

Les limites de cette métrique sont qu'elle est relative au nombre de domaines possibles à l'instant t , elle ne permet donc pas une comparaison aisée entre 2 générations de produits, faisant intervenir un nombre différent de domaines. De plus, les résultats de l'évaluation peuvent varier en fonction de celui qui l'a faite. Il faut donc faire cette évaluation avec la personne en charge du projet (qui a une vue d'ensemble) et quelques ingénieurs experts dans les domaines pour trancher.

3.3 Métriques d'innovation (MI)

3.3.1 Enjeux

Cet indicateur permet de mesurer le taux d'innovation d'un système mécatronique, permettant d'évaluer la « compétitivité » du système sur le marché. Aussi, nous nous intéressons à la répartition de cette innovation au sein du système. En effet, suivant l'organisation de l'entreprise, l'activité de conception mécatronique peut être distribuée sur des équipes « domaine » ou sur des équipes transversales autour d'un module, d'un sous-système... de nature pluridisciplinaire. Etant donné que l'aspect innovant d'un système est souvent plus lié à sa technologie qu'à une discipline, nous avons choisi de prendre en compte plutôt les services métiers s'appuyant sur les technologies, par exemple Energie/Alimentation, Capteurs, Traitement du signal, Mécanique/Mécanismes..., que sur les disciplines scientifiques (Mécanique, Electronique, Automatique, Informatique...). Nous les appellerons par la suite « domaines-métiers ». De la même manière, dans certains cas, les domaines de la mécatronique ne sont pas suffisants pour estimer le niveau de couplage entre les domaines (par exemple, l'énergie est un domaine important dans l'alimentation des systèmes mais il ne fait pas initialement partie des domaines de référence de la mécatronique que nous avons choisis). C'est pourquoi, nous pouvons facilement étendre l'indicateur d'intégration multi-domaine (MMDI) à un indicateur d'intégration des domaines « métier ». En effet, la définition de domaine peut dépendre de fait de l'organisation de l'entreprise et varier d'une entreprise à une autre, la définition de métiers, plus large, laisse la possibilité de réunir plusieurs domaines au sein d'un même département, tout en ayant les mêmes problématiques d'interfaçage avec les autres départements.

Aussi, pour définir la métrique d'innovation, nous déclinerons l'indicateur d'innovation suivant ces deux modes d'organisation. Ainsi, l'architecte système, en charge d'évaluer et de choisir une architecture, pourra au préalable avoir une vue du niveau d'innovation des éléments soit par service/équipe métier, soit par sous-système.

3.3.2 Métrique d'innovation (MI)

Cette métrique est composée d'un seul indicateur (l'Indicateur d'Innovation II_n) qui prend en compte le degré d'innovation pour chacun des domaines-métiers ou pour chaque élément d'un niveau d'abstraction n donné.

La métrique d'innovation est alors la somme des indicateurs pour chaque niveau de décomposition (sous-système, modules, composants), avec des coefficients de pondération, au choix du concepteur, pourvu que leur somme soit égale à 1. En effet, le concepteur peut rarement évaluer directement le taux d'innovation du système global à concevoir, c'est-à-dire au niveau d'abstraction « système », cette information résultera plutôt de la métrique d'innovation issue de l'ensemble des indicateurs des niveaux des sous-systèmes, modules et composants, plus faciles à estimer. Ainsi, pour calculer cette métrique, le concepteur doit être capable d'attribuer un « niveau d'innovation » sur l'ensemble des composants, modules

et sous-systèmes, car la propriété d'héritage des attributs des éléments entre deux niveaux successifs de la décomposition hiérarchique ne s'applique pas pour le degré d'innovation. En effet, à la différence des autres métriques, le degré d'innovation de l'élément « père » du niveau supérieur, n'est pas forcément égal à la « moyenne » des niveaux d'innovation des domaines métiers des éléments « fils » le composant. Effectivement, l'innovation d'un élément, au-delà de l'innovation intrinsèque de ses composants, peut également résulter de la manière d'intégrer ces composants. Ainsi, il est nécessaire de s'assurer que le degré d'innovation d'un élément du niveau est au moins égal au degré d'innovation maximal des éléments du niveau inférieur qui le constituent. Bien évidemment cet indicateur peut être calculé sur tous les niveaux de décomposition, la métrique d'innovation est alors la somme pondérée des indicateurs d'innovation de chaque niveau (la somme de ces coefficients λ_n étant égale à 1).

$$MI = \sum_{n=1}^{nl} \lambda_n II_n \quad (2.9)$$

nl : nombre de niveaux.

3.3.3 Indicateur d'innovation (II_n)

Pour cet indicateur, nous avons utilisé la fonction caractéristique \mathcal{I} ou encore appelée fonction indicatrice.

Si A est une partie de l'ensemble E , on définit la fonction indicatrice de A , qui est une application de E dans $\{0, 1\}$, notée \mathcal{I}_A , telle que :

$$\forall x \in E, \mathcal{I}_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{sinon} \end{cases} \quad (2.10)$$

Cela veut dire qu'elle nous indique si un point x appartient ou pas au sous-ensemble A . Elle nous donne 1 s'il lui appartient, 0 sinon.

Dans notre cas, nous voulons savoir si, pour un élément d'un niveau n , les domaines métiers liés à la conception mécatronique sont présents ou non dans cet élément dans le niveau n considéré.

La valeur de cette fonction vaut 1 lorsque le domaine étudié est présent dans l'élément considéré.

$$\mathcal{I}_{Me_i}(M_j) = \begin{cases} 1 & \text{si } M_j \in \bigcup_{e_i \in E} Me_i = \{M_j \in \mathcal{M} \mid \exists M_j \in \mathcal{M}, M_j \in Me_i\} \\ 0 & \text{si } M_j \notin \bigcup_{e_i \in E} Me_i \end{cases} \quad (2.11)$$

Avec :

$M = \{M_1, M_2, \dots, M_i, \dots, M_e\}$: l'ensemble des domaines métiers M_j de référence de l'entreprise impliqués dans la conception mécatronique

Me_i : ensemble des domaines-métiers de référence impliqués pour la conception de l'élément e_i d'un niveau d'abstraction n considéré.

i : indice de l'élément dans un niveau d'abstraction donné

j : indice du domaine métier considéré.

- ☞ Si l'activité de conception mécanique au sein de l'entreprise s'organise autour de domaines-métiers, l'indicateur d'innovation s'appuiera sur la somme de l'innovation associée à chaque service métier. Ainsi pour chaque domaine métier considéré M_j , la valeur d'innovation vaut :

$$II_{Mj} = \frac{\sum_{i=1}^{n_e} J_{Mei}(M_j) \times I_{ij}}{I_{max} \times n_e} \quad (2.12)$$

Avec :

n_M : nombre total de domaines métiers de référence de l'entreprise impliqués dans la conception mécanique parmi toutes les architectures étudiées.

n_e : nombre total d'éléments du niveau n

$J_{Mei}(M_j)$: fonction caractéristique du domaine métier M_j dans l'ensemble M_{e_i}

I_{ij} : valeur d'innovation du domaine métier M_j de l'élément e_i parmi l'ensemble des niveaux d'innovation suivants :

$I_{ij} = 1$: re-conception

$I_{ij} = 2$: re-conception + innovation

$I_{ij} = 3$: innovation complète

I_{max} : valeur maximale que peut prendre I (dans notre cas $I_{max} = 3$)

Dans ce premier cas, l'expression globale de l'indicateur d'innovation II_n au niveau d'abstraction n s'exprime par :

$$II_n = \frac{\sum_{j=1}^{n_M} II_{Mj}}{n_M} \quad (2.13)$$

- ☞ Si l'activité de conception mécanique au sein de l'entreprise s'organise transversalement autour des sous-systèmes pluridisciplinaires, l'indicateur d'innovation s'appuiera sur la somme de l'innovation associée à chaque élément de niveau d'abstraction n (sous-système ou module). Ainsi pour chaque élément considérée e_i , la valeur d'innovation vaut :

$$II_{ei} = \frac{\sum_{j=1}^{n_M} J_{Mei}(M_j) \times I_{ij}}{I_{max} \times n_M} \quad (2.14)$$

Dans le deuxième cas, la formule de l'indicateur d'innovation II_n devient :

$$II_n = \frac{\sum_{i=1}^{n_e} II_{ei}}{n_e} \quad (2.15)$$

La valeur de l'indicateur d'innovation II_n reste la même quelle que soit la vue (domaine-métier ou élément) relative à l'organisation de l'activité de conception mécanique, considérée, puisque les opérateurs « somme » utilisés sont commutatifs.

3.3.4 Bornes

Conformément aux spécifications que nous nous sommes fixées pour l'élaboration des métriques (Chapitre 1 §3.2.1) et notamment sur l'aspect normalisation, les bornes de cet indicateur sont 0 et 1. La valeur 1 est obtenue lorsque tous les domaines métiers impliqués dans la conception mécatronique sont complètement innovants et donc cotés à $I_{ij} = 3$ pour tous les éléments du niveau n considéré de l'architecture. La valeur 0 est obtenue lorsqu'aucun domaine métier de référence n'est utilisé.

3.3.5 Exemple

Pour illustrer cet indicateur, nous considérons le niveau module m dans un exemple théorique ci-dessous, les domaines-métiers sont notés Me pour Mécanisme, Ep pour Electronique de puissance, In pour Instrumentation (capteurs, mesure, traitement du signal...), Lo pour Logiciel et Co pour Communication (protocoles, IHM..), puis les valeurs d'innovation sont notées par un chiffre sur le côté droit du domaine-métier auquel il fait référence. Par exemple, pour un élément dont les domaines-métiers sont les mécanismes et l'électronique de puissance et qui ont respectivement un niveau d'innovation de 3 et 2, l'élément s'écrira Me3Ep2. Nous obtenons ainsi les résultats suivants :

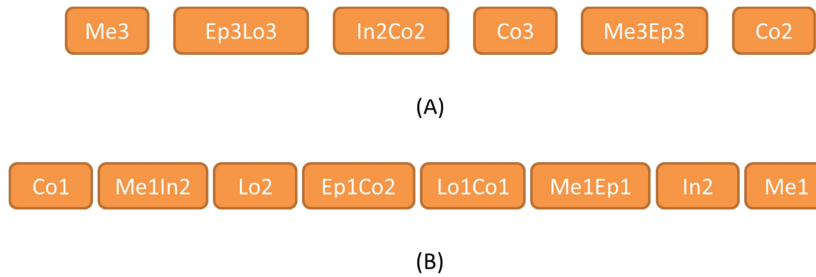


Figure 2-22 : Exemple théorique avec enrichissement des éléments en fonction de leur valeur d'innovation

Me : Mécanisme

Ep : Electronique de puissance

In : Instrumentation

Co : Communication

Lo : Algorithme, Logiciel

Pour le système (A)

- Indicateur d'innovation par domaine métier

Pour le domaine des mécanismes : $II_{mMeA} = \frac{1 \times 3 + 0 + 0 + 0 + 1 \times 3 + 0}{3 \times 6} = \frac{6}{18} = 0,5$

Pour le domaine de l'électronique de puissance : $II_{mEpA} = \frac{0 + 1 \times 3 + 0 + 0 + 1 \times 3 + 0}{3 \times 6} = \frac{6}{18} = 0,5$

Pour le domaine de l'instrumentation : $II_{mInA} = \frac{0 + 0 + 1 \times 2 + 0 + 0 + 0}{3 \times 6} = \frac{2}{18} \cong 0,1111$

Pour le domaine de la communication : $II_{mCoA} = \frac{0 + 0 + 1 \times 2 + 1 \times 3 + 0 + 1 \times 2}{3 \times 6} = \frac{7}{18} \cong 0,3889$

Pour le domaine du logiciel : $II_{mLoA} = \frac{0 + 1 \times 3 + 0 + 0 + 0 + 0}{3 \times 6} = \frac{3}{18} \cong 0,1667$

- Indicateur d'innovation par module

Pour le module 1 : $II_{m1A} = \frac{1 \times 3 + 0 + 0 + 0 + 0}{3 \times 5} = \frac{3}{15} = 0,2$

Pour le module 2 : $II_{m2A} = \frac{0 + 1 \times 3 + 0 + 0 + 1 \times 3}{3 \times 5} = \frac{6}{15} = 0,4$

Pour le module 3 : $II_{m3A} = \frac{0+0+1 \times 2 + 1 \times 2 + 0}{3 \times 5} = \frac{4}{15} \cong 0,2667$

Pour le module 4 : $II_{m4A} = \frac{0+0+0+1 \times 3 + 0}{3 \times 5} = \frac{3}{15} = 0,2$

Pour le module 5 : $II_{m5A} = \frac{1 \times 3 + 1 \times 3 + 0 + 0 + 0}{3 \times 5} = \frac{6}{15} = 0,4$

Pour le module 6 : $II_{m6A} = \frac{0+0+0+1 \times 2 + 0}{3 \times 5} = \frac{2}{15} \cong 0,1333$

Quelle que soit la vision de l'organisation des activités de conception, l'indicateur d'innovation pour ce niveau est alors égal à $II_m = \frac{\frac{6}{18} + \frac{6}{18} + \frac{2}{18} + \frac{7}{18} + \frac{3}{18}}{5} = \frac{\frac{3}{15} + \frac{6}{15} + \frac{4}{15} + \frac{3}{15} + \frac{6}{15} + \frac{2}{15}}{6} = \frac{12}{45} = 0,2667$

Pour le système (B)

- Indicateur d'innovation par domaine métier

Pour le domaine des mécanismes : $II_{mMeB} = \frac{0+1 \times 1 + 0 + 0 + 0 + 1 \times 1 + 0 + 1 \times 1}{3 \times 8} = \frac{3}{24} = 0,125$

Pour le domaine de l'électronique de puissance : $II_{mEpB} = \frac{0+0+0+1 \times 1 + 0 + 1 \times 1 + 0 + 0}{3 \times 8} = \frac{2}{24} \cong 0,0833$

Pour le domaine de l'instrumentation : $II_{mInB} = \frac{0+1 \times 2 + 0 + 0 + 0 + 0 + 1 \times 2 + 0}{3 \times 8} = \frac{4}{24} \cong 0,1667$

Pour le domaine de la communication : $II_{mCoB} = \frac{1 \times 1 + 0 + 0 + 1 \times 2 + 1 \times 1 + 0 + 0 + 0}{3 \times 8} = \frac{4}{24} \cong 0,1667$

Pour le domaine du logiciel : $II_{mLoB} = \frac{0+0+1 \times 2 + 0 + 1 \times 1 + 0 + 0 + 0}{3 \times 8} = \frac{3}{24} = 0,125$

- Indicateur d'innovation par module

Pour le module 1 : $II_{m1B} = \frac{0+0+0+1 \times 1 + 0}{3 \times 5} = \frac{1}{15} \cong 0,0667$

Pour le module 2 : $II_{m2B} = \frac{1 \times 1 + 0 + 1 \times 2 + 0 + 0}{3 \times 5} = \frac{3}{15} = 0,2$

Pour le module 3 : $II_{m3B} = \frac{0+0+0+0+1 \times 2}{3 \times 5} = \frac{2}{15} \cong 0,1333$

Pour le module 4 : $II_{m4B} = \frac{0+1 \times 1 + 0 + 1 \times 2 + 0}{3 \times 5} = \frac{3}{15} = 0,2$

Pour le module 5 : $II_{m5B} = \frac{0+0+0+1 \times 1 + 1 \times 1}{3 \times 5} = \frac{2}{15} \cong 0,1333$

Pour le module 6 : $II_{m6B} = \frac{1 \times 1 + 1 \times 1 + 0 + 0 + 0}{3 \times 5} = \frac{2}{15} \cong 0,1333$

Pour le module 7 : $II_{m7B} = \frac{0+0+1 \times 2 + 0 + 0}{3 \times 5} = \frac{2}{15} \cong 0,1333$

Pour le module 8 : $II_{m8B} = \frac{1 \times 1 + 0 + 0 + 0 + 0}{3 \times 5} = \frac{1}{15} \cong 0,0667$

L'indicateur d'innovation pour ce niveau est alors égal à $II_m = \frac{\frac{3}{24} + \frac{2}{24} + \frac{4}{24} + \frac{4}{24} + \frac{3}{24}}{5} = \frac{\frac{1}{15} + \frac{3}{15} + \frac{2}{15} + \frac{3}{15} + \frac{2}{15} + \frac{2}{15} + \frac{2}{15} + \frac{1}{15}}{8} = \frac{2}{15} \cong 0,1333$

3.3.6 Résultats et discussion

Ces résultats montrent que le module 5 du système A est celui qui présente l'innovation la plus élevée, tandis que le domaine-métier le plus impacté par l'innovation est celui de la communication. Concernant le système B, l'innovation est plus présente dans les modules 2 et 4, et dans le domaine-métier des mécanismes.

En effet, cet indicateur d'innovation associé à un niveau d'abstraction donné, permet une mesure relative du taux d'innovation au sein d'un système (le taux est en effet ramené au

nombre d'éléments de l'architecture étudiée), elle mesure l'effort d'innovation à un instant t pour une architecture donnée. Ainsi, utilisée sur deux générations de produits, elle indiquera l'effort lié à l'innovation de chacune des architectures par rapport à l'existant considéré à l'instant t de chaque génération pour ne pas fausser la comparaison et prendre en compte l'évolution technologique réalisée entre les deux.

3.4 Métrique de Mise en Œuvre d'un Projet Mécatronique (MMOPM)

Aujourd'hui, les coûts associés à la mise en œuvre d'un projet mécatronique dépendent fortement des enjeux définis dans les paragraphes précédents, à savoir l'intégration multi-niveau, l'intégration multi-domaine et l'innovation du système à concevoir. Aussi pour en évaluer quantitativement et donc maîtriser en amont l'effort, les ressources (techniques et humaines) et le coût associés, nous proposons une Métrique de Mise en Œuvre d'un Projet Mécatronique (MMOPM). Contrairement aux deux dernières métriques, qui permettaient d'évaluer un aspect (multi-domaine, d'innovation) relatif à l'architecture étudiée (pour un nombre d'éléments donné pour un niveau d'abstraction donné), mais ne permettaient pas de comparer quantitativement l'effort de développement associé à cet aspect des différentes architectures étudiées, la MMOPM doit permettre une mesure absolue de ces aspects permettant de comparer quantitativement le coût associé à chacune des architectures.

Pour cela, nous devons prendre en compte dans la normalisation de l'indicateur, le nombre d'éléments maximal pour un niveau d'abstraction donné parmi toutes les architectures. Cela suppose que les métriques ne peuvent être calculées qu'à un instant donné pour un ensemble donné d'architectures candidates et à chaque nouvelle génération de produit ou ajout d'architecture candidate, il faut recalculer le nombre maximum de domaines- métiers et le nombre maximum d'éléments.

La métrique d'intégration multi-domaine absolue $MMDI'$ est donc normalisée par le nombre maximal d'éléments pour chaque niveau de toutes les architectures confondues et devient :

$$MMDI' = \sum_{j=1}^{nd} \lambda_j IMD'_j \quad (2.16)$$

Avec

$$IMD'_j = \frac{\frac{1}{nd} \sum_{i=1}^{n_e} card(D_i)}{n_{e_{max}}} \quad (2.17)$$

$n_{e_{max}}$ étant le nombre maximum d'éléments pour un niveau d'abstraction considéré parmi toutes les architectures.

La métrique d'innovation absolue MI' est normalisée par le nombre maximum de domaines-métiers de référence impliqués dans les éléments d'un niveau, toutes architectures confondues et par le nombre maximal d'éléments de chaque niveau de toutes les architectures confondues, elle devient :

$$MI' = \sum_{n=1}^{nl} \lambda_n II'_n$$

✎ Avec pour l'indicateur d'innovation absolu par domaine métier :

$$II'_{Mj} = \frac{\sum_{i=1}^{n_e} J_{Mei}(M_j) \times I_{ij}}{I_{max} \times n_{e_{max}}} \quad (2.18)$$

$n_{e_{max}}$ étant le nombre maximum d'éléments pour un niveau d'abstraction considéré parmi toutes les architectures

L'indicateur d'innovation absolu devient :

$$II'_n = \frac{\sum_{j=1}^{n_M} II'_{Mj}}{n_{M_{max}}} \quad (2.19)$$

$n_{M_{max}}$ étant le nombre maximum de domaines-métiers impliqués dans les éléments parmi toutes les architectures.

✎ De même pour l'indicateur d'innovation absolu pour chaque élément e_i d'un niveau donné, il est défini par :

$$II'_{ei} = \frac{\sum_{j=1}^{n_M} J_{Mei}(M_j) \times I_{ij}}{I_{max} \times n_{M_{max}}} \quad (2.20)$$

L'indicateur d'innovation absolu peut alors s'écrire :

$$II'_n = \frac{\sum_{i=1}^{n_e} II'_{ei}}{n_{e_{max}}} \quad (2.21)$$

La Métrique de Mise en Œuvre d'un Projet mécatronique, peut alors s'exprimer comme la somme pondérée et normalisée des trois métriques :

$$MMOPM = \lambda_1 MMLI + \lambda_2 MMDI' + \lambda_3 MI' \quad (2.22)$$

Cette métrique bornée entre 0 et 1 pourra, suivant l'organisation de l'entreprise, être multipliée par une valeur de référence :

- de coût, pour estimer le coût d'un projet, le coût de développement d'un composant, en fonction de son degré d'innovation, de son intégration multi-domaine et multi-niveau
- d'hommes-mois, pour évaluer tantôt la charge de l'architecte système, tantôt les ressources nécessaires pour chaque équipe métier « sous-système » ou « domaine »,
- de mois de formation des personnels, en fonction de l'aspect multi-domaine du système
- de mois, pour prévoir la durée de développement du système
- ...

En effet, suivant que le projet fait intervenir de nombreuses équipes techniques associées (et donc domaines/domaines métiers), les problématiques de l'intégration multi-domaine interviendront de façon prépondérante dans la valeur de cette métrique pour quantifier l'effort de développement, le coût, les besoins de formations et la durée du projet.

Mais cette métrique prend en compte de la même manière le degré d'innovation du système pour évaluer l'effort de développement du produit dans chaque équipe technique ou chaque sous-système.

La métrique MMOPM est enfin couplée également avec la métrique d'intégration multi-niveau pour évaluer le niveau d'intégration du système et notamment à quel niveau d'abstraction du système, l'effort d'intégration sera le plus important et donc quel est

l'acteur qui devra fournir le plus d' « effort ». Ainsi, l'effort de développement incombera aux concepteurs spécialistes « monodomaine » au niveau composant, aux équipes techniques « métier » pour le niveau module et sous-système et à l'architecte système pour le niveau système.

L'intérêt industriel de coupler toutes ces métriques au sein d'une seule métrique est de pouvoir évaluer et donc maîtriser en amont le temps total d'heures de travail à allouer (et donc le planning et le coût associés) pour la mise en œuvre d'un produit mécatronique : que ce soit dans le nombre d'équipes techniques associées (si on considère une équipe par domaine/module) (métrique d'intégration multi-domaine) et donc des interfaçages nécessaires correspondants, que dans l'effort d'innovation/reconception (métrique d'innovation), que dans l'intégration multi-niveau (métrique d'intégration multi-niveau) qui impacte directement les problématiques de l'intégration physique.

4 Expérimentation

Nous avons réalisé une expérimentation sur un système mécatronique intégré avec plusieurs architectures existantes pour leur appliquer les métriques multi-niveaux et multi-domaine. Malheureusement, par manque de retours industriels sur l'effort d'innovation, qui reste une donnée assez sensible, nous n'avons pas pu expérimenter la métrique de mise en œuvre d'un projet mécatronique sur ces systèmes réels.

4.1 Description du système

Le système mécatronique que nous avons choisi pour illustrer les deux premières métriques d'intégration est celui du système « moteur-roue », dont nous donnerons les spécifications dans le paragraphe suivant. Puis nous détaillerons 3 architectures candidates du système pour lesquelles nous appliquerons les précédentes métriques, afin de les comparer selon l'objectif d'intégration multi-niveau et multi-domaine.

4.1.1 *Un système mécatronique intégré*

Le choix du système moteur-roue, comme illustration d'un système mécatronique intégré vient de ses différents niveaux d'intégration.

En effet, si on reprend les 3 types d'intégration précédemment définis (Chapitre 1 §2.1) : fonctionnelle, multi-domaine et physique, nous trouvons dans le moteur-roue des éléments correspondant à chacune d'elle.

Intégration fonctionnelle :

Le système moteur-roue permet d'intégrer la fonction de motorisation mais également de support à la propulsion/traction électrique de véhicules légers, et parfois même la suspension et le freinage.

Intégration multi-domaine :

Dans les différentes architectures de moteur-roue que nous allons présenter, plusieurs domaines et notamment ceux de la mécatronique sont intégrés : le domaine de la mécanique pour la motorisation et pour le support du système, celui de l'électrique/électronique pour la gestion de la puissance du moteur... De plus certaines fonctions comme la gestion de l'alimentation, le contrôle/commande du système à distance sans fil permet d'intégrer des domaines comme l'automatisme et l'informatique.

Intégration physique :

Concernant l'intégration physique du système, Espanet (Espanet 1999a) fait une synthèse des spécifications géométriques à prendre en compte pour assurer l'aptitude du système à être intégré dans une roue. Le moteur doit avoir une largeur restreinte pour pouvoir être couvert totalement par la jante et le pneumatique. Cela implique que le moteur soit inversé, c'est-à-dire que le rotor soit à l'extérieur pour faciliter le montage de celui-ci directement sur la jante. On voit bien que la compacité du système est alors déterminée par le volume enveloppe de la roue, avec en plus les contraintes dynamiques (dues à la rotation de la roue) sur l'ensemble des sous-systèmes.

L'idée d'intégrer un moteur dans une roue d'une voiture n'est pas récente. En 1868, Ferdinand Porsche développa le premier prototype de moteur-roue. Le principe est simple, il s'agissait d'un moteur électrique dont le stator est solidaire de l'essieu et le rotor est fixé à la jante.

L'avantage de ce système est de réduire l'encombrement et ainsi libérer de la place sous le capot pour par exemple placer les batteries. Le deuxième avantage est l'absence de transmission qui diminue aussi les pertes par frottement.

Nous ne présentons pas une veille technologique exhaustive sur les structures existantes des systèmes « moteur-roue », mais un choix de 3 types d'architecture pour illustrer l'utilisation de nos métriques dans le choix d'architecture lors de la conception de systèmes mécatroniques intégrés:

- le moteur-roue ez-Wheel mis au point par la société du même nom³⁴,
- le moteur-roue utilisé dans l'Irisbus Cristalis, conçu et fabriqué par Alstom³⁵
- et l'Active Wheel développée par Michelin³⁶.

4.1.2 Modélisation hiérarchique des différentes architectures candidates du système

Nous avons utilisé le langage SysML pour modéliser les différentes architectures hiérarchiques à l'aide de diagrammes de définition de blocs (bdd). Cette modélisation a été effectuée à l'aide de l'outil Artisan Studio version 7.4 d'Atego³⁷. Le BDD est un diagramme qui permet de définir la structure du système et sa décomposition hiérarchique à l'aide de liens de composition. Un bloc peut représenter n'importe quel élément d'un niveau d'abstraction donné. Un lien de composition indique que l'élément fait physiquement partie d'un autre élément (d'un niveau d'abstraction plus élevé).

4.1.2.1 Hypothèses/Cahier des charges

Pour pouvoir être cohérent, nous avons besoin de comparer des architectures candidates qui répondent au même cahier des charges. Or les 3 systèmes existants industriels n'ont pas exactement les mêmes fonctionnalités. Aussi, nous avons décidé de choisir un cahier des charges « fictif » qui permettrait d'inclure chacune des architectures existantes de ces systèmes, comme tout ou partie d'une des architectures candidates de notre système.

La liste des fonctions auxquelles devra répondre notre système est la suivante :

³⁴ <http://ez-wheel.com/>

³⁵ http://fr.wikipedia.org/wiki/Irisbus_Cristalis

³⁶

http://www.mobilisconference.com/uploads/2012/pdf/Michelin_Recherche_Technique_DP%20Mobilis_Nov%202012_FR.pdf

³⁷ <http://www.atego.com/products/atego-modeler/>

- F1 : s'intégrer physiquement dans une roue dont le pneumatique aura un diamètre standard
- F2 : permettre la traction électrique d'un véhicule.
- F3 : permettre également le freinage du véhicule,
- F4 : recevoir et traduire les commandes de l'utilisateur
- F5 : maintenir le contact avec un sol irrégulier (suspension).
- F6 : être alimenté et gérer le niveau de batterie.

Lorsque les architectures des systèmes existants n'intégraient pas l'ensemble de ces fonctions, nous avons fait le choix de les matérialiser par des technologies (inconnues) typées « Hors-domaine », partant du principe qu'elles n'impacteraient pas ou peu les enjeux d'intégration/couplage évaluées par les métriques proposées, puisqu'elles devraient être rajoutées par juxtaposition au système déjà existant.

4.1.2.2 Le moteur-roue ez-Wheel³⁸



Mis au point en 2009 par la société du même nom, l'ez-Wheel est un moteur-roue électrique autonome en énergie. Le moyeu de la roue intègre tous les organes d'une chaîne de traction électrique, c'est-à-dire un moteur électrique de traction, une batterie et une électronique de puissance permettant de contrôler le moteur et de gérer la batterie. De plus, ce système possède une interface de commande sans-fil, mais n'intègre pas de système de suspension, ni de système de freinage. Ce système est principalement dédié

aux applications légères comme une trottinette, une brouette, un fauteuil roulant, un chariot...

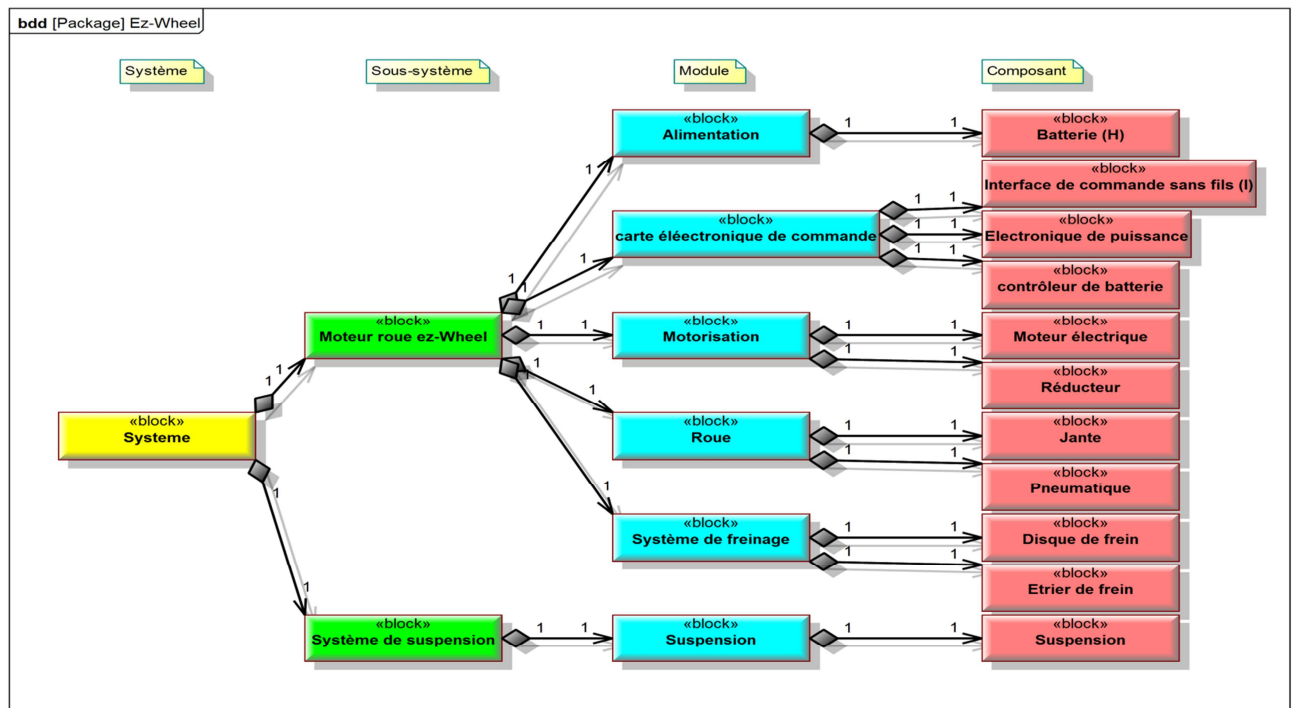


Figure 2-23 : Modélisation hiérarchique du système ez-Wheel

³⁸ <http://ez-wheel.com/>

4.1.2.3 Le moteur-roue produit par GEC Alsthom

Le Cristalis est un trolleybus fabriqué par Irisbus (Espanet 1999a) et équipé d'une chaîne de traction électrique construite par Gec Alsthom. La particularité de ce trolleybus est l'intégration de moteur-roue électrique dans les roues arrière. Ce système est alimenté par un coffret trolley (hors du système) permettant de capter le courant des lignes aériennes de contact. La difficulté d'installer un moteur de traction de 80 kW dans le moyeu de la roue les a contraints à concevoir un moteur ultra-compact tournant à très haute vitesse (9 000 tr/min), refroidi par eau. C'est un moteur triphasé asynchrone choisi pour son excellent rapport poids/puissance. Ce moteur est couplé à un réducteur sur lequel vient se fixer la jante. Pour réaliser un ensemble le plus compact possible, un pneu double-largeur développé par Michelin permet de remplacer les classiques pneus jumelés. Un frein à bain d'huile complète l'ensemble.

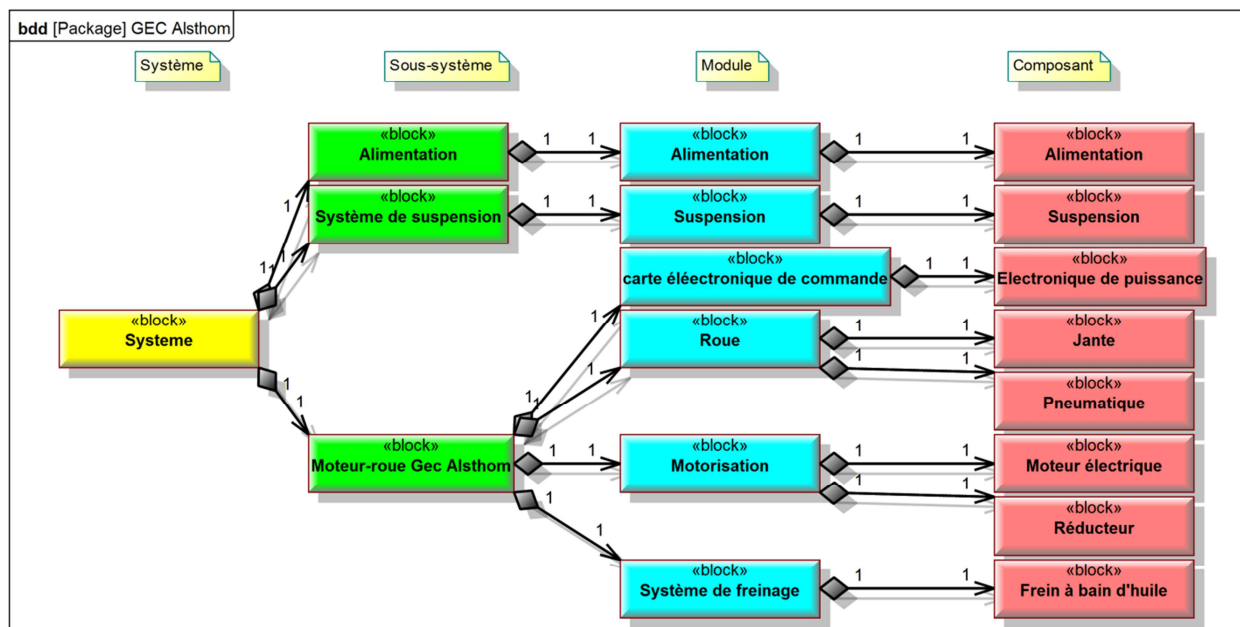


Figure 2-24 : Moteur-roue créé par Alstom pour l'Irisbus Cristalis

4.1.2.4 Le moteur-roue Active Wheel de chez Michelin³⁹



Comparé aux autres systèmes de moteur-roue, l'Active Wheel développé par Michelin comprend, en plus des fonctions de motorisation et de freinage, un moteur et un système de suspension active électrique qui permet de faire varier la hauteur suivant les obstacles détectés et l'inclinaison de la caisse dans les virages pour améliorer la tenue de route. L'inconvénient de ce système est la complexité mécanique, la réduction de la maintenabilité et la fiabilité.

³⁹ <http://fr.wikipedia.org/wiki/Moteur-roue>

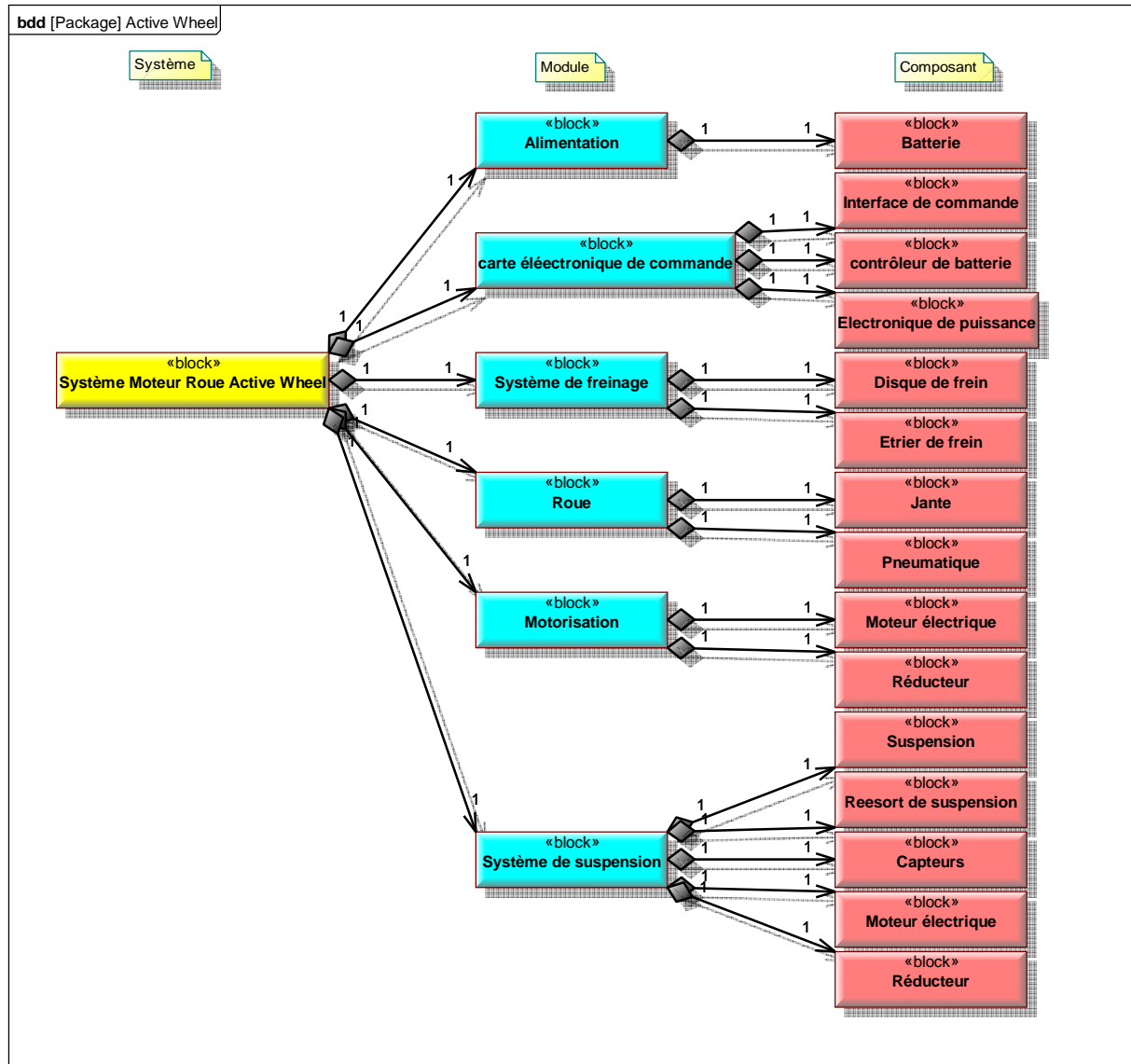


Figure 2-25: Modélisation hiérarchique du moteur-roue "Active Wheel" de chez Michelin

4.2 Résultats

4.2.1 Calcul de la métrique d'intégration multi-niveau : MMLI

Nous calculons la métrique d'intégration multi-niveau pour les trois architectures candidates.

Pour cela, nous avons calculé les 3 indicateurs d'intégration multi-niveau au niveau local (IML₁, IML₂, IML₃) et l'indicateur d'intégration multi-niveau au niveau global (IML₅). Les résultats sont donnés dans le tableau ci-dessous (Table 2-1).

Indicateur d'intégration multi-niveau (IML)		ez-Wheel	GEC Alsthom	Active Wheel
Niveau Global	ILMS	5/33 ≈0,15	1/12 ≈0,08	1/2 ≈0,50
Niveau Système/Sous-système	IML1	1/2 ≈0,50	2/3 ≈0,67	
Niveau Sous-Système/Module	IML2	2/5 ≈0,40	1/4 ≈0,25	5/6 ≈0,83
Niveau Module/composant	IML3	13/36 ≈0,36	1/6 ≈0,17	89/180 ≈0,49

Table 2-1 : Résultats des indicateurs d'intégration multi-niveau

Pour calculer la métrique d'intégration multi-niveau nous avons choisi un coefficient de pondération de $\frac{1}{2}$ pour l'indicateur global, les autres coefficients valent $\frac{1}{6}$ pour les systèmes ez-Wheel et Gec et $\frac{1}{4}$ pour le système Active Wheel qui ne présente que deux niveaux d'intégration au vu de sa très forte intégration. Voici le résultat du calcul de la métrique d'intégration multi-niveau (Table 2-2) :

	ez-Wheel	GEC Alsthalm	Active Wheel
Métrique d'intégration multi-niveau	179/626 $\approx 0,29$	2/9 $\approx 0,22$	419/720 $\approx 0,58$

Table 2-2 : Résultat de la métrique d'intégration multi-niveau

4.2.2 Calcul de la métrique d'intégration multi-domaine : MMDI

Nous avons enrichi les éléments des trois architectures avec leur domaine, soit M pour Mécanique, E pour Electronique, A pour Automatique, I pour Informatique et H pour hors-domaine de la mécatronique. Les architectures sont les suivantes :

- Pour la première architecture, utilisant un système ez-Wheel (Figure 2-26) :

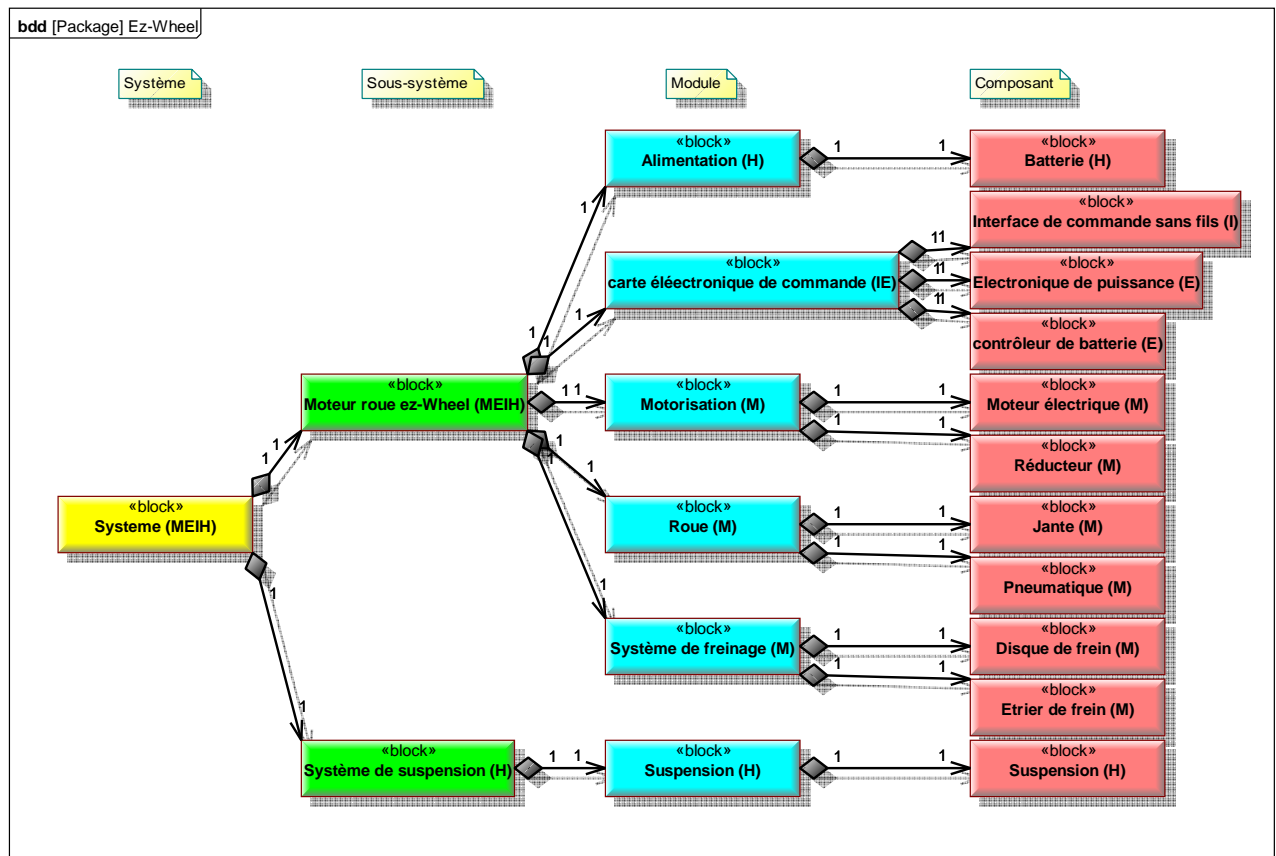


Figure 2-26 : Décomposition par domaine pour l'architecture utilisant le système ez-Wheel

- Pour la deuxième architecture utilisant le système Gec (Figure 2-27) :

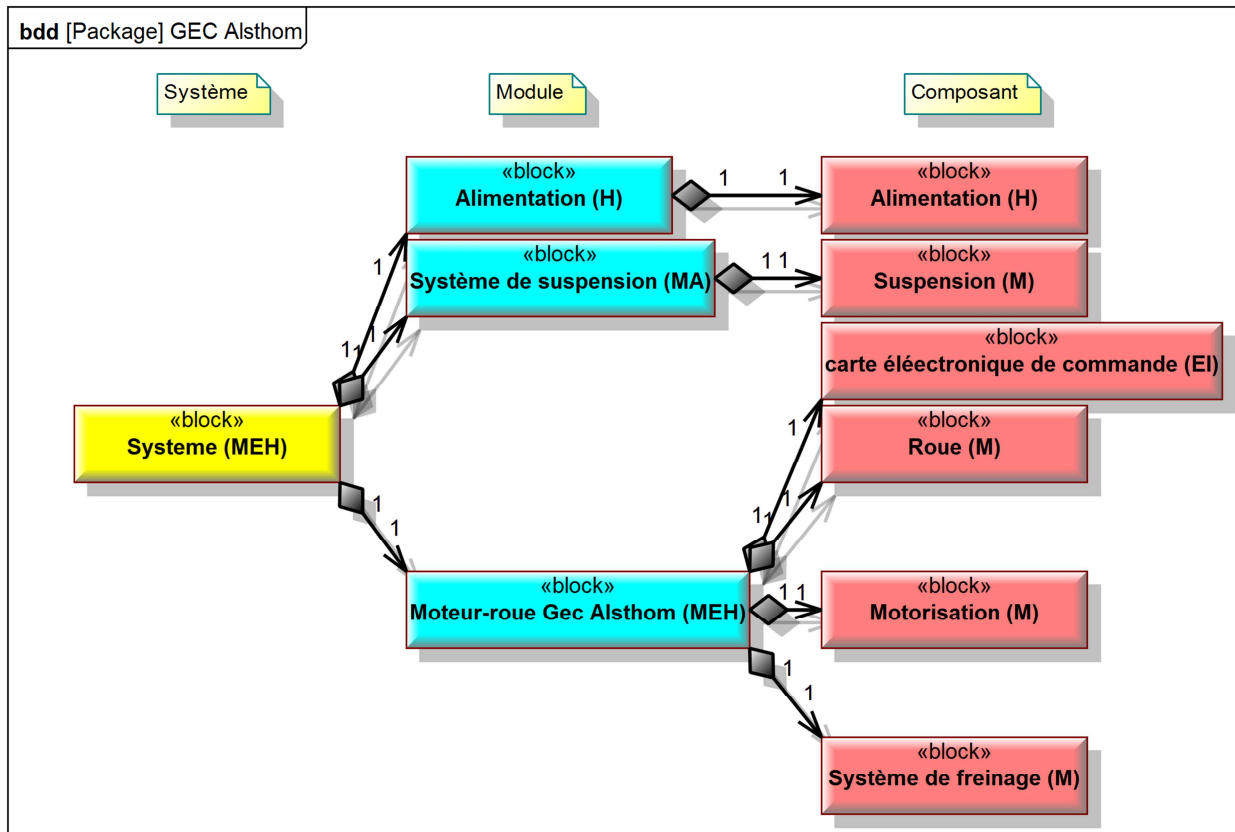


Figure 2-27 : Décomposition par domaine pour l'architecture utilisant le système Gec

Dans cette architecture il n'y a que 3 niveaux d'abstraction considérés pour la métrique multi-domaine, puisque le 3^{ème} niveau est déjà composé d'éléments mono-domaines.

- Pour finir, l'architecture intégrant le système Active Wheel (Figure 2-28) :

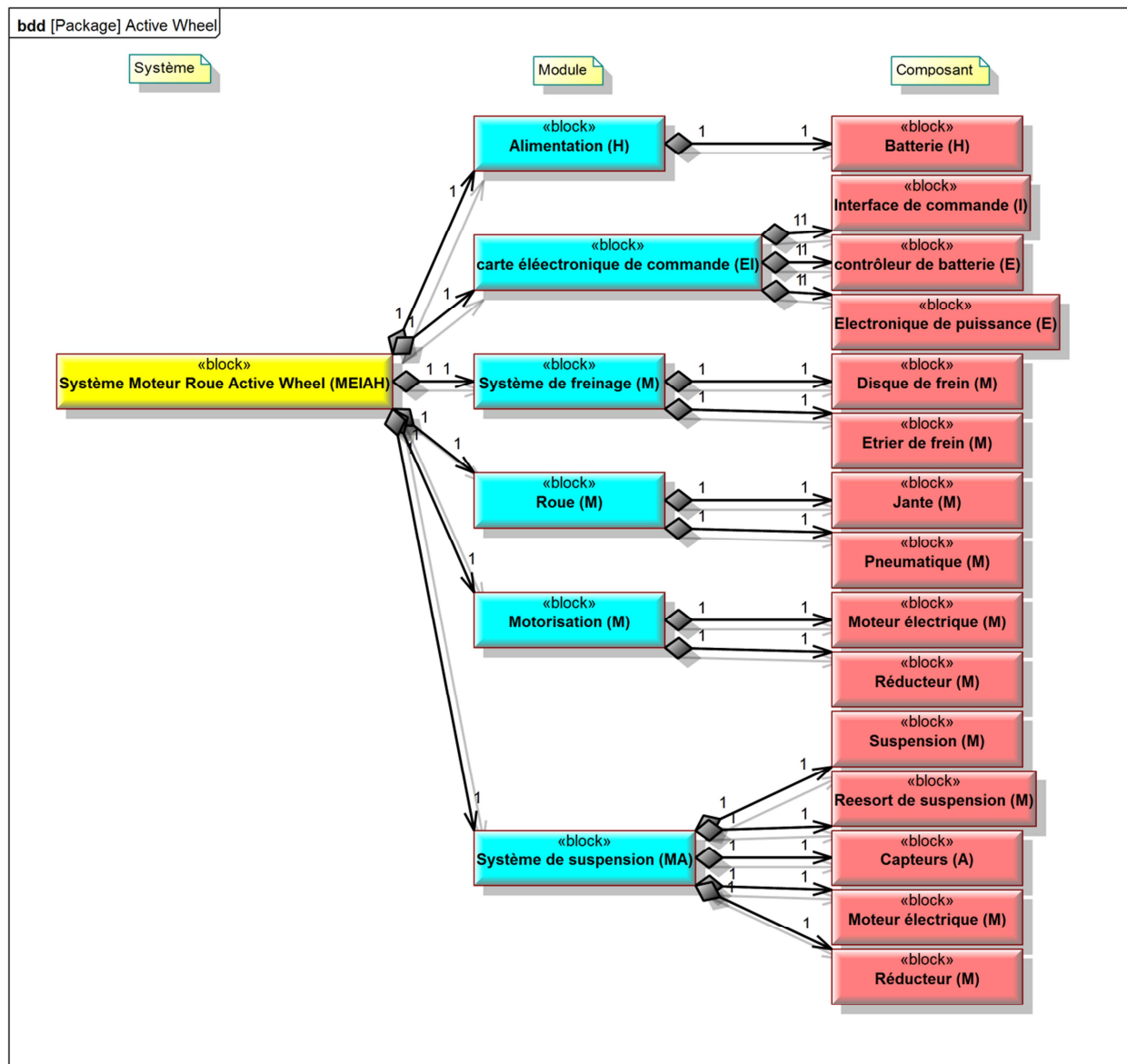


Figure 2-28 : Décomposition par domaine pour l'architecture utilisant le système Active Wheel

Pour calculer la métrique d'intégration multi-domaine mécatronique, nous avons évalué au préalable les indicateurs d'intégration multi-domaine IMD pour chaque niveau d'abstraction. Les résultats des calculs sont donnés dans le tableau récapitulatif suivant (Table 2-3) :

Indicateur Multi-domaine		ez-Wheel	GEC Alsthom	Active Wheel
Niveau Système/global	IMD1	3/4 =0,75	1/2 =0,5	1 =1
Niveau Sous-système	IMD2	3/8 =0,375		
Niveau Module	IMD3	5/24 ≈0,21	1/6 ≈0,167	7/24 ≈0,292
Niveau Composant	IMD4	9/44 ≈0,205	1/8 =0,125	7/30 ≈0,233

Table 2-3 : Résultat des indicateurs d'intégration multi-domaine IMD pour chaque niveau d'abstraction

Pour le calcul, nous avons choisi arbitrairement les coefficients de pondération λ_i de la manière suivante: nous avons choisi un coefficient de $\frac{1}{2}$ pour l'indicateur IMD_1 , qui

représente l'indicateur global, puis pour l'ensemble des coefficients d'indicateurs locaux, nous avons choisi le coefficient $1/6$ pour le système ez-Wheel, et $1/4$ pour les systèmes Gec et Active Wheel qui n'ont que 3 niveaux d'abstraction pris en compte pour l'intégration multi-domaine, afin de conserver une somme des coefficients égale à 1.

Les résultats du calcul de cette métrique sont donnés dans le tableau ci-dessous (Table 2-4) :

	ez-Wheel	GEC Alsthom	Active Wheel
Métrique d'Intégration Multi-Domaine	401/792 $\approx 0,51$	31/96 $\approx 0,32$	101/160 $\approx 0,63$

Table 2-4 : Résultats du calcul de la métrique d'intégration multi-domaine mécatronique

4.3 Analyse des résultats et discussions

Concernant l'intégration multi-niveau, nous avons schématisé les indicateurs d'intégration multi-niveau pour chaque niveau d'abstraction et pour chaque architecture dans le graphique ci-dessous (Figure 2-29) :

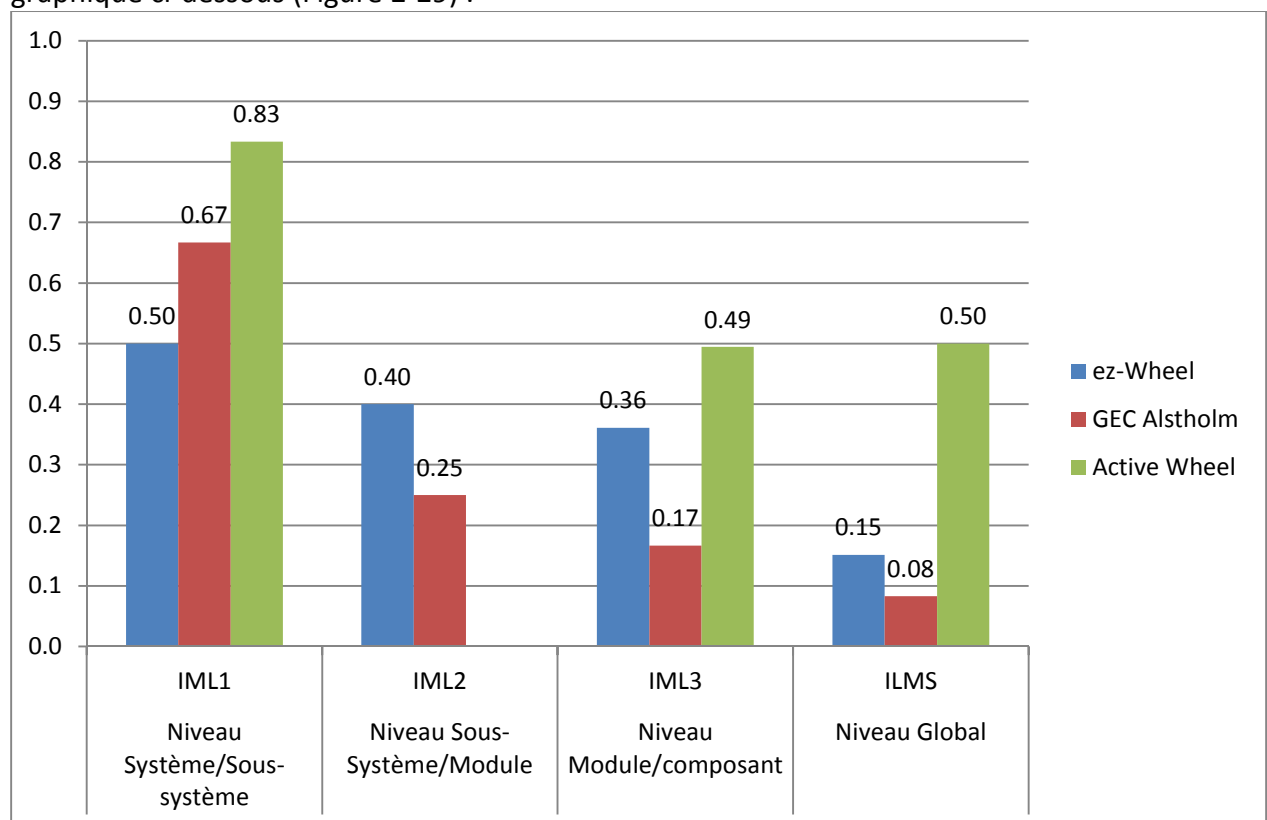


Figure 2-29 : Récapitulatif des indicateurs d'intégration multi-niveau pour chaque architecture et pour chaque niveau d'abstraction

Nous pouvons voir qu'il n'y a pas d'indicateur IML_2 pour le système Active Wheel. En effet, ce système est intégré sur seulement 3 niveaux d'abstraction. De plus, c'est le système qui a le plus de composants c'est pourquoi l'indicateur d'intégration global est bien plus élevé que pour les deux autres systèmes.

Avec à peu près le même nombre de composants pour les systèmes ez-Wheel et GEC (respectivement 11 et 8), nous pouvons constater qu'il y a un fort effort d'intégration au niveau système/sous-système pour le GEC alors que l'effort d'intégration est plus régulier pour le système ez-Wheel.

Par ailleurs, nous avons résumé les différents résultats des indicateurs d'intégration multi-domaine (IMD) pour chaque architecture et chaque niveau d'abstraction dans le graphique ci-dessous (Figure 2-30) :

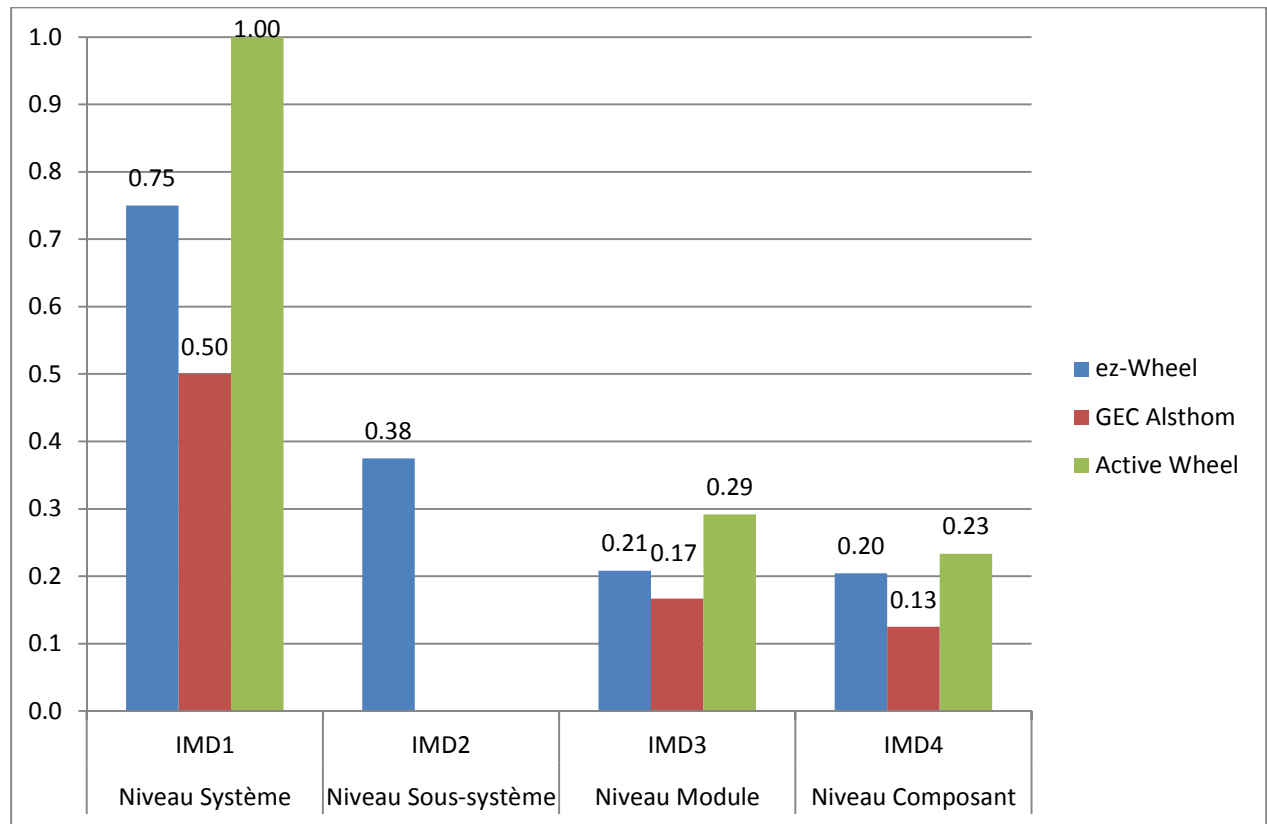


Figure 2-30: Résultat des indicateurs d'intégration multi-domaine pour chaque architecture et pour chaque niveau d'abstraction

Premièrement, il ne subsiste pas d'indicateur d'intégration multi-domaine au niveau des sous-systèmes IMD_2 pour les systèmes utilisant le GEC d'Alsthom et l'Active Wheel de Michelin. En effet, puisqu'il manque un niveau de décomposition, il n'existe pas de niveau sous-système pour ces deux systèmes.

Le moteur-roue Active Wheel est seul à réunir les quatre domaines de référence de la mécanique : son indicateur IMD_1 qui est l'indicateur global a la valeur maximale de 1. Cependant, il intègre d'autres domaines que ceux de la mécanique puisque l'indicateur IMD_4 (qui est celui des composants) n'atteint pas la valeur maximale de 0,25.

Le moteur-roue GEC quant à lui ne contient seulement que deux domaines de référence de la mécanique puisque l'indicateur IMD_1 est égale à 0,5. De plus, il comprend aussi des domaines ne faisant pas partie des domaines de référence que nous avons définis, puisque l'indicateur IMD_4 n'atteint pas non plus la valeur maximale de 0,25. Nous pouvons constater également que ce système a les indicateurs les plus bas des 3 systèmes candidats.

Pour finir, la structure utilisant le système ez-Wheel comporte 3 des 4 domaines de la mécanique, son indicateur IMD_1 vaut 0,75. Comme les deux autres systèmes, d'autres domaines de la mécanique sont présents puisque l'indicateur IMD_4 est inférieur à 0,25.

A présent, nous allons comparer les différentes architectures et donc les différentes solutions à l'aide des métriques d'intégration multi-domaine et multi-niveau. Le récapitulatif des résultats de l'ensemble de ces métriques est donné par le tableau ci-après (Table 2-5) :

	ez-Wheel	GEC Alsthom	Active Wheel
Métrique d'intégration Multi-niveau	179/626 ≈0,29	2/9 ≈0,22	419/720 ≈0,58
Métrique d'intégration Multi-domaine	401/792 ≈0,51	31/96 ≈0,32	101/160 ≈0,63

Table 2-5 : Tableau récapitulatif des résultats des métrique d'intégration multi-domaine et multi-niveau

Nous avons synthétisé les résultats dans un graphique (Figure 2-31) qui a pour abscisse la métrique d'intégration multi-domaine et pour ordonnée la métrique d'intégration multi-niveau.

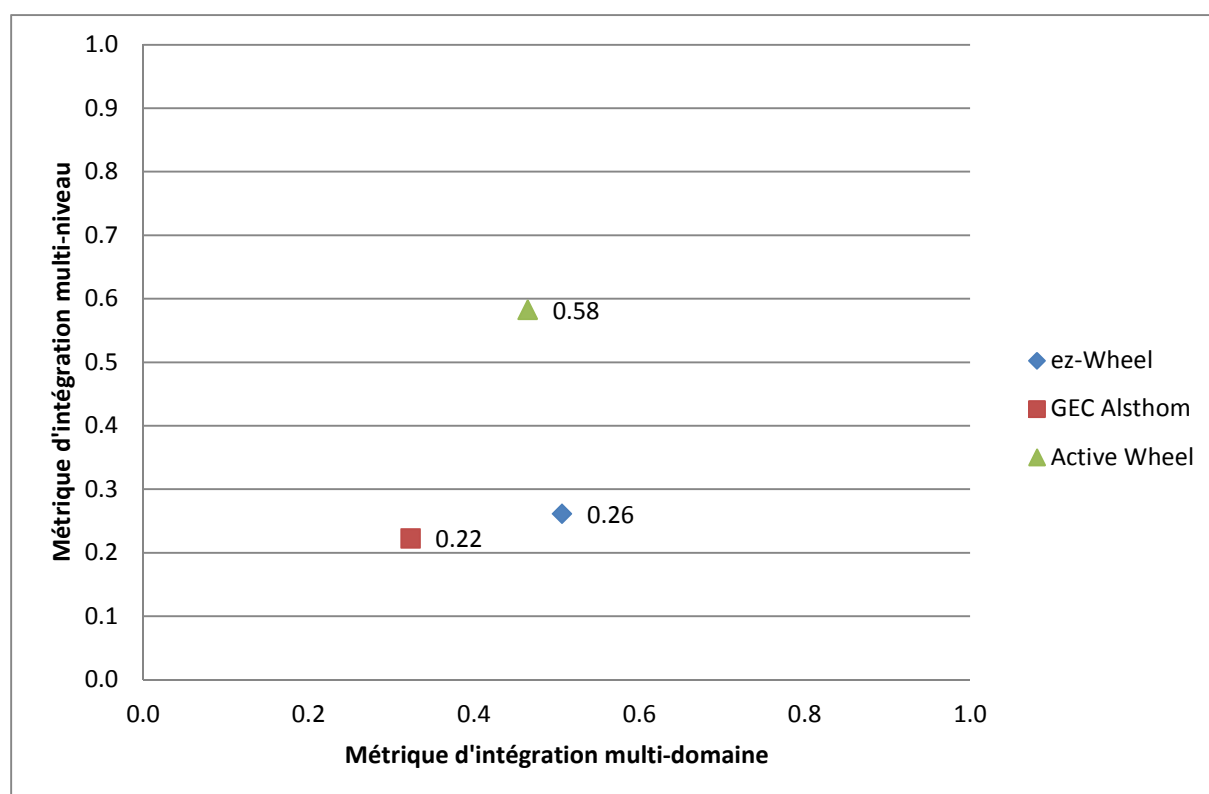


Figure 2-31 : Analyse des solutions candidates à l'aide des métriques d'intégration multi-domaine et multi-niveau

Sur le graphique, nous pouvons constater que le système GEC Alsthom est celui qui est le moins intégré au niveau multi-domaine et au niveau multi-niveau. Au contraire, le système Active Wheel de Michelin est la solution la plus intégrée aussi bien pour l'intégration multi-domaine que pour l'intégration multi-niveau. Ce graphique permet d'avoir une vue d'ensemble du niveau d'intégration des différents systèmes.

Le choix final d'avoir une architecture solution plus intégrée ou non revient au concepteur. Les avantages d'une architecture intégrée sur l'aspect « multi-domaine » sont : un travail collaboratif entre les équipes plus important, une meilleure optimisation globale, la réduction de volume ... Cependant, une architecture intégrée sur l'aspect « multi-niveau » diminue l'autonomie de la structure, sa modularité, et l'accessibilité, la maintenance et le recyclage sont plus difficiles.

5 Discussions

Il existe trois principales difficultés dans l'établissement de ces métriques.

Pondération des indicateurs

La première concerne la pondération de ces indicateurs ou à plus haut niveau de différentes métriques (ex pour MMOPM) nécessaire si l'on souhaite une évaluation globale.

Comme il est difficile de trouver un critère unique prenant en compte la multiplicité des aspects relatifs à tout contexte de prise de décision, l'approche multicritère permet de construire, en étroite collaboration avec le décideur, une solution satisfaisante qui résulte du meilleur compromis (Roy 2005).

On peut identifier deux catégories d'approches multicritères: l'une qui conserve des indicateurs distincts et une autre qui a tendance à synthétiser l'évaluation en prenant une seule mesure représentant les indicateurs de manière synthétique.

- Approche des indicateurs distincts:

Cette vision tient la spécificité de chaque mesure (James L Mathieson 2010), et permet d'associer un vecteur (avec ses n indicateurs appartenant à R_n), à l'entité considérée (ici l'architecture). Une représentation graphique couramment utilisée est le schéma en étoile avec un axe par indicateur (Figure 2-32).

Par conséquent, il est impossible avec cette approche, de classer deux objets selon un objectif global.

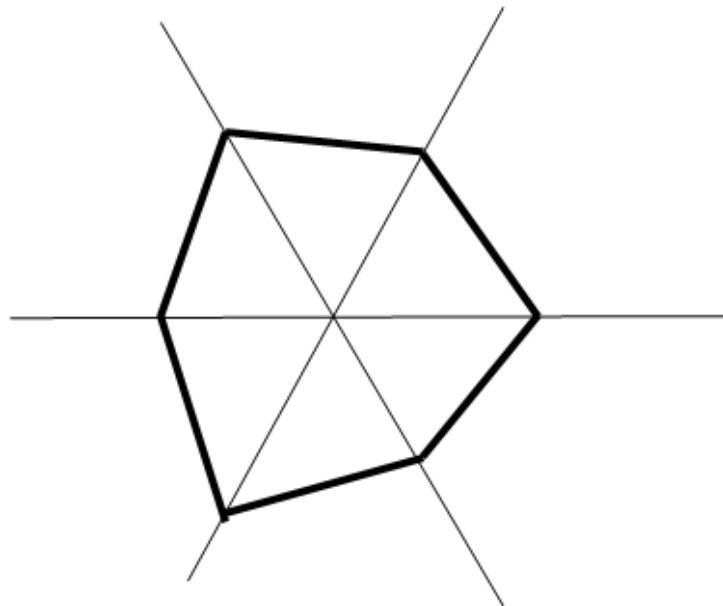


Figure 2-32 : Exemple de diagramme en étoiles associé à six indicateurs.

- Approches avec des indicateurs intégrés dans une seule métrique.

Ce type d'approche offre une métrique numérique synthétique qui sera associée à l'objet, d'une manière simple et sans ambiguïté, de sorte que la comparaison et la classification des objets puissent être effectives.

Dans le contexte de notre évaluation de l'intégration mécatronique dans l'espace à trois dimensions, la synthèse de plusieurs indicateurs abstraits (multi-niveaux, multi-domaines) unifiés en une seule métrique semblait présenter certains avantages.

Dans notre approche, la dimension de l'espace métrique est 4, que ce soit pour MMLI ou MMDI. La représentation graphique des objets est donc limitée à des projections dans les plans à cet espace R^4 .

Un critère général unique pour synthétiser une métrique M pour un objet "a", basée sur n indicateurs, dont les valeurs mesurables sont comprises entre 0 et 1, peut être exprimé comme:

$$M(a) = F(I_1(a), I_2(a), \dots, I_n(a)) \quad (2.23)$$

Cette métrique permet ainsi d'avoir des ordres de grandeur comparables pour différents objets.

La fonction F peut prendre les formes suivantes (Ehrgott 2005):

- Formes algébriques avec opérateur multiplicateur sur $I_n(a)$,
- Distances statistiques proches du formalisme des plans d'expériences,
- Agrégations additives,

$$M(a) = \sum_{j=1}^4 \lambda_j v_j I_j(a) \quad (2.24)$$

- Ou enfin somme pondérée

$$M(a) = \sum_{j=1}^4 \lambda_j I_j(a) \quad (2.25)$$

Après avoir évalué l'applicabilité de ces différentes formes, nous avons choisi d'utiliser la somme pondérée. En effet, c'est un modèle simple adapté à l'utilisation en contexte technique et industriel, du fait de sa facilité d'utilisation.

Pourtant, certaines limites de la somme pondérée ont été identifiées. Dans cette approche, les critères peuvent être considérés comme compensatoires, et dans notre contexte, l'interprétation de ces compensations peut ne pas être intuitive. De plus, la sensibilité de toute classification utilisant la somme pondérée vis-à-vis des valeurs de coefficients de pondération, peut conduire à un ordre complètement différent des résultats même avec de petites variations des poids choisis (Ehrgott 2005). Par conséquent, il conviendra de faire une analyse de sensibilité, comme nous le recommandons dans notre processus de construction de métriques décrit au Chapitre 4 (Warniez et al. 2014A), pour étudier comment la métrique varie lorsque certains paramètres évoluent. Enfin, une analyse de la robustesse devra être également réalisée avec de vrais cas de test industriels, afin de tester si une mesure jugée satisfaisante à un instant t, le reste sous différents scénarios et avec plusieurs jeux de données.

Choix de la prise en compte de plusieurs critères pour l'aide à la décision

De nombreuses études portent sur la prise en compte de plusieurs critères, dont les principes, avantages et inconvénients sont largement détaillées dans certaines synthèses bibliographiques (Figueira et al. 2005; Bouyssou 2006). Les trois grandes familles de l'analyse multicritère relative à la méthode d'agrégation (Moullec et al. 2015) sont les suivantes:

- méthodes d'agrégation complète : MAUT (Multi-Attribute Utility Theory), AHP (Analytic Hierarchy Process) ou MACBETH (Measuring Attractiveness by a Categorical Based Evaluation Technique) ;
- méthodes d'agrégation par surclassement (agrégation partielle) : ELECTRE (élimination Et Choix Traduisant la Réalité) et PROMETHEE (Préférence Classement Organisation méthode d'enrichissement Evaluations)
- et les méthodes d'agrégation locaux : STEM (méthode pas à pas)).

Par ailleurs, d'autres approches permettent de traiter les cas où les critères sont contradictoires et gênent la prise de décision (Rudolph 1995; Couturier et al. 2014).

Cependant dans l'industrie, bien qu'il existent de bonnes pratiques et des outils pour la conception de l'architecture du système, il devient également nécessaire de prendre en compte les méthodes de travail de collaboration spécifiques ainsi que l'implication des différents départements de conception du système dans la prise de décision finale du choix d'architecture (Ben Hamida et al. 2015b).

Evaluation subjective

Enfin, des études ont montré le risque de dérive de subjectivité lorsque plusieurs individus étaient impliqués dans un processus collaboratif pour déterminer une valeur commune (Arduin et al. 2013), et notamment dans des approches faisant intervenir des facteurs de pondération (Ittner et al. 2003).

Pour limiter ce risque, notamment dans les entreprises où de nombreuses parties prenantes interviennent dans la conception, il est recommandé de déterminer préalablement à la mesure effective de la métrique sur plusieurs architectures candidates, de rassembler les avis de toutes les personnes impliquées dans la conception et de s'appuyer sur le retour d'expérience des précédentes conceptions, pour déterminer de façon consensuelle si possible et officielle les facteurs de pondérations qui seront appliqués pour le choix d'architecture. Dans la pratique, cette tâche peut être difficile, car elle exige beaucoup de temps à capitaliser et analyser toutes ces données.

Un deuxième aspect concerne la durée de vie d'une métrique, en effet si une métrique doit être utilisée pendant une longue période (sur des décennies ou des plusieurs générations de produits), les facteurs de pondération peuvent évoluer en fonction de la stratégie de l'entreprise.

Par ailleurs, lorsque l'on considère les métriques d'innovation ou des métriques de processus, un changement significatif (de la société, la technologie ...) peut même remettre en question la pertinence de toute la métrique. Un produit donné, même dans la même gamme de produits, peut être si différent que la métrique ne puisse pas permettre d'évaluer les nouveaux produits avec les mêmes paramètres. Typiquement, si l'on observe la différence entre les premiers disjoncteurs (dont la fonction était principalement de protéger de façon automatique un circuit électrique contre les dommages causés par une surcharge ou un court-circuit) et les disjoncteurs actuels, intelligents et connectés (qui sont entièrement intégrés aux maisons intelligentes, permettant de contrôler la consommation d'énergie, la régulation de l'énergie, etc.), il est évident dans ce contexte que l'objectif de la modularité du produit ou de l'intégration de produit et les métriques correspondantes ne peuvent pas être considérés de la même manière. Avec les progrès de la discipline mécatronique, les attentes des clients et les défis techniques ont évolué si rapidement -et encore plus vite avec la révolution informatique et l'émergence de nouveaux systèmes cyber-physiques- que les métriques basées sur les compétences, les méthodes, les outils des

concepteurs actuels ne pourront pas demeurer les mêmes, car elles ne prendront pas en compte les évolutions économiques, environnementaux, sociaux et technologiques.

Ainsi, il est recommandé de considérer qu'une métrique a une durée de vie, qui détermine la période de temps pendant laquelle nous la supposons valide.

Lorsqu'une évolution de métrique est envisagée, il faut garantir sa mise à jour, par sa validation au travers de nouveaux cas-tests industriels (voir Chapitre 4).

6 Automatisation du calcul des métriques

L'automatisation des métriques d'intégration multi-niveau et multi-domaine a été faite par export des modèles SysML aux formats XMI et XML, dans le logiciel Artisan Studio de la société Atego, puis passage avant de procéder au calcul sous Excel.

Du logiciel Artisan Studio nous avons pu extraire les données des différents diagrammes au moyen de modèles SysML exportés au format XML et XMI. Préalablement, dans le modèle SysML, pour chaque niveau d'architecture (Système, Sous-système, Module et Composant), un diagramme de définition des blocs est défini avec ses différents éléments (blocs) constituant le niveau considéré et les liens de composition correspondants entre le système et les différents éléments pour chaque niveau.

L'automatisation consiste à « lire et parser » les deux fichiers exportés pour chaque modélisation d'architecture candidate pour extraire les données utiles au calcul de nos métriques dans un fichier Excel.

Le premier fichier (au format XML) permet d'identifier le numéro de chaque élément et permet également d'identifier les liens de composition. Les éléments doivent être référencés par la première lettre du niveau (S : System, SS : Sub-System, M : Module, C : Component) puis par un numéro. La Figure 2-33 montre un exemple d'identification des blocs avec leur lettre et leur numéro d'identification ID.

	A	B	C	D
1				
2		Lecture		
3				
4		Blocks déclarés dans le chapitre "Activities"		
5				
6		Fichier :	MMLI_Proto_MultiArchi_V1.xml	
7				
8		Block	ID	
9		C1	0a8cdfcb-d838-4212-a1c8-5196b2bfb234	
10		C2	eecac2bf-6eb1-477c-ad33-c3fad2772633	
11		C3	3457a200-c72c-4bd2-8385-d8dbbd31517f	
12		C4	cc0a1588-4dde-4a6f-9067-6117fd2525b0	
13		C5	4e8b3d79-770e-472a-8ada-2702dee2bbd4	
14		C6	c990264c-30de-4c04-a3a5-b866910d0571	
15		C7	6783c8df-b79d-4b0e-9f43-98c34168d771	
16		M1	f30d3d16-1fac-47c0-952c-884e799534a4	
17		M2	b3cf8b82-42b7-4f80-895f-190b383d7671	
18		M3		
19		M4	01f987e6-5a78-48cc-ba33-300b32236c24	
20		M5	c20a05da-165e-4ebe-99c0-de277857f569	
21		SS1	0bb33df1-7e4c-4270-97bf-dd7363eef80a	
22		SS2	f120d9a5-43a0-4739-aaa7-16645d6eaeff	
23		SS3	6d15b7e6-959e-4c3e-8c63-3abf6118ee8c	
24		S	8ac93115-ad60-4b4f-8b6d-f46c70d037f8	

Figure 2-33 : Exemple d'éléments des architectures pour chaque niveau

Puis, toujours grâce au fichier XML, nous pouvons identifier les différents numéros d'identification des diagrammes de définition des blocs permettant de modéliser les éléments pour chaque niveau d'abstraction. La Figure 2-34 donne un exemple des identifications de chaque diagramme pour le niveau système, sous-système, module et composant ainsi que pour les trois diagrammes de composition.

	A	B	C
2		Lecture	
3			
4		Blocks déclarés dans le chapitre "Block Definition Diagrams"	
5			
6		Fichier :	MML_Proto_MultiArchi_V1.xml
7			
8		Block	ID
9		C1	0a8cdfcb-d838-4212-a1c8-5196b2bfb234
10		C2	eeccac2bf-6eb1-477c-ad33-c3fad2772633
11		C3	3457a200-c72c-4bd2-8385-d8dbbd31517f
12		C4	cc0a1588-4dde-4a6f-9067-6117fd2525b0
13		C5	4e8b3d79-770e-472a-8ada-2702dee2bbd4
14		C6	c990264c-30de-4c04-a3a5-b866910d0571
15		C7	6783c8df-b79d-4b0e-9f43-98c34168d771
16		M1	f30d3d16-1fac-47c0-952c-884e799534a4
17		M2	b3cf8b82-42b7-4f80-895f-190b383d7671
18		M3	
19		M4	01f987e6-5a78-48cc-ba33-300b32236c24
20		M5	c20a05da-165e-4ebe-99c0-de277857f569
21		SS1	0bb33df1-7e4c-4270-97bf-dd7363eef80a
22		SS2	f120d9a5-43a0-4739-aaa7-16645d6eaeff
23		SS3	6d15b7e6-959e-4c3e-8c63-3abf6118ee8c
24		S	8ac93115-ad60-4b4f-8b6d-f46c70d037f8
25			
26		Diagrammes	ID
27		NiveauSystème	51210717-b513-41a3-91d7-d718149e5f7f
28		NiveauSous-Système1	c01bfc49-5cbd-45f1-a463-3ecd1295f060
29		NiveauSous-Système2	c68e21db-18c9-4710-b30b-d605691bbd86
30		NiveauSous-Système3	c26f35me-06g6-1803-w95w-c458732f25
31		NiveauModule1	48c37ge7-gbe9-74a6-3abb-d671583v6bfd
32		NiveauModule2	34b14fg6-ga7-36f6-2ffc-b328841a6bbj
33		NiveauModule3	85d7c9-df5c-4ba1-a15e-6dfb386f6f18
34		NiveauComposant1	56a87cc3-afc6-56f7-4ceg-a862453c6fac
35		NiveauComposant2	33f74bb1-cba3-45d4-8abf-b994740e9baf
36		NiveauComposant3	38f6f151-bf0a-4ba1-a15e-6dfb386f6f18
37		DiagrammeComposition1	3f9b08f7-0678-420a-9afe-982ae493e24f
38		DiagrammeComposition2	785b9e4b-e55e-47e7-8ede-0dc1b859ccb5
39		DiagrammeComposition3	f7b811fc-007f-4947-86fe-b03c665540f7
40			

Figure 2-34 : Identifiant des éléments de chaque diagramme et identifiants des différents digrammes

Le deuxième fichier qui est un fichier au format xmi, permet d'identifier les liens de composition qui lient les différents éléments entre chaque niveau pour chaque architecture donnée.

La Figure 2-35 permet de sélectionner les fichiers XML et xmi ainsi que les diagrammes de composition des différentes architectures. Dans notre exemple, il y a 3 architectures candidates. Puis le bouton « Calcul des indicateurs » permet, comme son nom l'indique, de calculer les indicateurs IML_1 , IML_2 , IML_3 et ILM_5 afin de pouvoir évaluer la métrique d'intégration multi-niveau pour ces 3 architectures.

	A	B	C	D
1				
2				
3		Calcul des indicateurs		RAZ
4				
5		Fichier XML	C:\MMLI_Proto_MultiArchi_V1.xml	...
6		Fichier XML	C:\MMLI_Proto_MultiArchi_V1.xml	...
7				
8		Diagrammes à analyser	Indicateurs	
9		DiagrammeAllocations1		
10		DiagrammeAllocations2		
11		DiagrammeAllocations3		
12				

Figure 2-35 : En-tête du fichier

Les figures suivantes affichent les résultats pour chaque architecture pour les indicateurs d'intégration multi-niveau.

	A	B	C	D
13				
14		DiagrammeComposition1		
15				
16		Blocs systèmes	Nombre de liens	$1-(1/nal_ek)$
17		S	2	0,5
18			IML1	0,5
19				
20		Blocs sous-systèmes	Nombre de liens	$1-(1/nal_ek)$
21		SS1	5	0,4
22		SS2	1	0
23			IML2	0,4
24				
25		Blocs modules	Nombre de liens	$1-(1/nal_ek)$
26		M1	1	0
27		M2	3	0,6666667
28		M3	2	0,5
29		M4	2	0,5
30		M5	2	0,5
31		M6	1	0
32			IML3	0,361111111
33				
34			IMLS	0,1515151
35			Nombre de composants	11
36			Nombre de modules	6
37			Nombre de sous-systèmes	2

Figure 2-36 : Calcul des indicateurs pour l'architecture 1

	A	B	C	D
39				
40		DiagrammeComposition2		
41				
42		Blocs systèmes	Nombre de liens	1-(1/nal_ek)
43		S	3	0,6666667
44			IML1	0,6666667
45				
46		Blocs sous-systèmes	Nombre de liens	1-(1/nal_ek)
47		SS1	4	0,75
48		SS2	1	0
49		SS3	1	0
50			IML2	0,25
51				
52		Blocs modules	Nombre de liens	1-(1/nal_ek)
53		M1	1	0
54		M2	1	0
55		M3	2	0,5
56		M4	2	0,5
57		M5	1	0
58		M6	1	0
59			IML3	0,1666667
60				
61			IMLS	0,08333333
62			Nombre de composants	8
63			Nombre de modules	6
64			Nombre de composants de sous-systèmes	3

Figure 2-37: Calcul des indicateurs pour l'architecture 2

	A	B	C	D
66				
67		DiagrammeComposition3		
68				
69		Blocs systèmes	Nombre de liens	1-(1/nal_ek)
70		S	6	0,8333333
71			IML1	0,83333333
72				
73		Blocs modules	Nombre de liens	1-(1/nal_ek)
74		M1	2	0,5
75		M2	4	0,75
76		M3	3	0,6666667
77		M4	1	0
78		M5	2	0,5
79		M6	2	0,5
80			IML3	0,49444444
81				
82			IML_S	0,5
83			Nombre de composants	15
84			Nombre de modules	6
85			Nombre de systèmes	1
86				

Figure 2-38 : Calcul des indicateurs pour l'architecture 3

Pour implémenter la métrique d'intégration multi-domaine, il faut stéréotyper chaque composant avec le domaine de la mécanique concerné. La solution envisagée est de stéréotyper les blocs avec 5 attributs/propriétés booléennes supplémentaires (1 pour chaque domaine de la mécanique, et 1 pour les autres domaines) et le concepteur indique vrai ou faux lorsque le domaine est présent dans le composant. Avec les liens de composition, les différentes propriétés sont remontées aux éléments du niveau supérieur.

Les 3 figures suivantes montrent les résultats de calcul automatique pour les indicateurs d'intégration multi-domaine.

A	B	C	D
3			
4	DiagrammeComposition1		
5			
6	Blocs systèmes	Nombre de domaine	$1/nd*card(D)$
7	S	3	0,75
8		IMD1	0,75
9			
10	Blocs sous-systèmes	Nombre de domaine	$1/nd*card(D)$
11	SS	3	0,75
12		IMD2	0,375
13			
14	Blocs modules	Nombre de domaine	$1/nd*card(D)$
15	M	5	1,25
16		IMD3	0,208333333
17			
18	Blocs composant	Nombre de domaine	$1/nd*card(D)$
19	C	9	2,25
20		IMD4	0,2045454
21			
22		Nombre de composants	11
23		Nombre de modules	6
24		Nombre de sous-systèmes	2

Figure 2-39 : Calcul des indicateurs d'intégration multi-domaine pour l'architecture 1

A	B	C	D
35			
36	DiagrammeComposition2		
37			
38	Blocs systèmes	Nombre de domaine	$1/nd*card(D)$
39	S	2	0,5
40		IMD1	0,5
41			
42	Blocs sous-systèmes	Nombre de domaine	$1/nd*card(D)$
43	SS	2	0,5
44		IMD2	0,166666667
45			
46	Blocs modules	Nombre de domaine	$1/nd*card(D)$
47	M	3	0,75
48		IMD3	0,125
49			
50		Nombre de composants	8
51		Nombre de modules	6
52		Nombre de sous-systèmes	3

Figure 2-40 : Calcul des indicateurs d'intégration multi-domaine pour l'architecture 2

A	B	C	D
53			
54			
55	DiagrammeComposition3		
56			
57	Blocs systèmes	Nombre de domaine	$1/nd*card(D)$
58	S	4	1
59		IMD1	1
60			
61	Blocs sous-systèmes	Nombre de domaine	$1/nd*card(D)$
62	SS	7	1,75
63		IMD2	0,291666667
64			
65	Blocs modules	Nombre de domaine	$1/nd*card(D)$
66	M	14	3,5
67		IMD3	0,233333333
68			
69		Nombre de composants	
70		Nombre de modules	15
71		Nombre de sous-systèmes	6

Figure 2-41 : Calcul des indicateurs d'intégration multi-domaine pour l'architecture 3

7 Conclusions et perspectives

Dans ce chapitre, nous nous sommes intéressés à la conception et au cycle de vie d'architectures des systèmes mécatroniques, et plus particulièrement aux métriques permettant au concepteur de faire ses choix d'architectures en fonction de ses objectifs.

Pour cela, nous avons proposé trois métriques, adressant respectivement l'intégration multi-niveau, l'intégration multi-domaine et l'effort d'innovation des systèmes mécatroniques. Après avoir développé les enjeux spécifiques de ces métriques pour le concepteur ou le chef de projet de systèmes mécatroniques, qui doit choisir une architecture parmi plusieurs architectures candidates, nous avons présenté les différentes métriques et leur calcul, avant de les expérimenter sur l'exemple d'un système intégrant un moteur-roue et assurant les mêmes fonctions initiales, à partir d'architectures proposées par 3 fabricants différents. Ces métriques permettent notamment de quantifier, pour chaque architecture l'effort de collaboration/interfaçage requis entre les quatre domaines d'expertise de la mécatronique (Automatique, Mécanique, Electronique et Informatique), et autant de métiers concernés, mais également entre les services d'intégration successive. Elles permettent ainsi de présenter une aide à la décision objective, connus préalablement de tous, qui réduira les temps de discussion et de développement inutile lors du choix d'une architecture. Enfin une dernière partie présente le développement de l'automatisation des calculs à partir de modèles d'architecture en SysML vers un fichier Excel.

Ces contributions adressent des métriques pour l'évaluation de l'effort d'intégration dans les architectures de systèmes mécatroniques. Aujourd'hui, il existe de nombreux autres objectifs tout aussi importants de considérer pour la conception mécatronique comme la sûreté, la robustesse, la fiabilité, l'utilité, la flexibilité qu'il faudra prendre en compte dans les perspectives de ces travaux, ainsi que l'étude de leurs couplages.

Chapitre 3

Métrique géométrique pour l'intégration physique

1	Introduction.....	123
1.1	Le contexte	123
1.2	Problématique et plan.....	123
2	Etat de l'art	124
2.1	Intégration physique des systèmes mécatroniques	124
2.2	Modélisation géométrique et placement des composants	130
2.3	Evaluation des architectures spatiales.....	132
2.4	Problématique.....	137
3	Métrique d'intégration physique développée	137
3.1	Indicateurs de compacité	139
3.2	Indicateur d'accessibilité	143
3.3	Métrique d'intégration physique (MIP)	145
4	Exemple mécatronique	146
4.1	Présentation du système et des différentes architectures.....	147
4.1	Mesure de la compacité réelle.....	149
4.2	Indicateur de forme.....	153
4.3	Indicateur de compacité convexe	154
4.4	Indicateur d'accessibilité	154
5	Implémentation des données géométriques en SysML.....	155
5.1	SysML et les profils.....	156
5.2	Définition de l'extension SysML pour la géométrie	156
5.3	Applications aux architectures spatiales de la motorisation hybride.....	159
5.4	Placement des composants.....	161
6	Discussion/perspectives	162
7	Conclusion	162

1 Introduction

1.1 Le contexte

Durant la phase de préconception, les spécifications des clients sont traduites en spécifications techniques par le biais de l'analyse fonctionnelle, et les architectures de système (fonctionnelles, logiques et physiques) sont générées. Ces activités sont généralement accomplies par l'architecte système (Chapitre 2 §2.2), qui évalue également les architectures candidates afin de sélectionner celle qui correspond le mieux aux spécifications des clients, avant de s'engager dans des analyses plus détaillées et souvent coûteuses (Hammadi 2012).

Habituellement la phase d'évaluation a lieu au cours de la phase de conception préliminaire. Durant cette phase, les équipes de simulation estiment les valeurs de performance des paramètres de conception en utilisant des outils de simulation ou des feuilles de calcul personnalisées de chaque concepteur (Chapitre 2 §2.3.4.5). Dès lors, la phase de conception préliminaire doit permettre d'aider à la prise de décision sur la faisabilité d'une architecture de système donnée avant d'entreprendre la conception détaillée. Les résultats de cette phase sont des modèles de simulation des différentes architectures proposées avec une quantification préliminaire des paramètres de conception critiques pour le choix de l'architecture à conserver.

Finalement, une fois l'architecture du système choisie, la phase de conception détaillée conduit à affiner les paramètres géométriques et physiques (de performance ou émergeant des interactions physiques : thermiques, électromagnétiques...) des composants, en utilisant la modélisation 3D et les simulations par éléments finis.

Rappelons que la conception mécatronique concerne les systèmes intégrant la mécanique, l'électronique, l'informatique et l'automatique dans un objectif d'intégration fonctionnelle, multi-domaine et physique. Elle vise à concevoir des systèmes plus compacts qui ont, généralement, l'avantage de réduire également leur poids. Par conséquent, la compacité devient un critère d'optimisation décisif pour les systèmes mécatroniques. Or, cette réduction de volume impose une attention particulière sur le placement des composants pour respecter l'enveloppe totale englobant le système. Seule cette vigilance est donc garante de compacité, et permet de restreindre les couplages multi-physiques indésirables liés à cette proximité des composants multi-domaines.

1.2 Problématique et plan

Aujourd'hui, le placement des composants a généralement lieu en phase de conception détaillée. Or comme l'objectif d'intégration physique est fondamental pour les systèmes mécatroniques, il est important de prendre en compte dès la phase de préconception la géométrie des composants et l'architecture spatiale de ces derniers, afin de pouvoir anticiper et limiter les risques inhérents à une position inadéquate des composants trop tardive en conception détaillée.

Dans ce chapitre, nous proposons de développer des métriques d'intégration physique, favorisant la prise en compte des contraintes géométriques et de positionnement, et cela dès la phase de préconception. L'objectif visé étant d'aider les concepteurs à sélectionner une architecture spatiale compatible avec l'objectif d'intégration physique des produits mécatroniques.

Ce chapitre est tout d'abord consacré à un état de l'art sur la définition de l'intégration physique, les méthodes existantes de modélisation géométrique et de placement des composants ainsi que les différentes mesures existantes de distances géométriques. Puis cette section se termine par les méthodes et métriques existantes pour évaluer l'intégration physique.

La section suivante présente nos travaux sur les métriques d'intégration physique pour l'évaluation des architectures spatiales candidates.

La troisième partie s'articule autour d'un exemple de système mécatronique, notamment sur différentes architectures d'une motorisation hybride, pour tester et analyser les indicateurs de la métrique d'intégration physique.

Ce chapitre est complété par le développement d'une extension du langage SysML permettant la prise en compte des caractéristiques géométriques des composants dès la description du « modèle système » au début de la conception.

Nous terminons ce chapitre sur une discussion de nos travaux et des perspectives envisagées, avant de conclure.

2 Etat de l'art

2.1 Intégration physique des systèmes mécatroniques

2.1.1 Définition

Selon la norme NF E01-010 (NF 2008), qui définit le vocabulaire associé à la mécatronique, un produit mécatronique est un produit disposant de la capacité de percevoir son milieu environnant, de traiter l'information, de communiquer et d'agir sur son milieu, tout en présentant un niveau complet d'intégration mécatronique, du point de vue fonctionnel et physique.

Par ailleurs rappelons que l'*intégration fonctionnelle* correspond à l'apport de fonctions de détection, de communication, de traitement de l'information et de rétroaction aux fonctions mécaniques de base. L'*intégration physique* répond à l'interpénétration des supports mécanique et électronique (qui embarquent les fonctions de l'automatique et l'informatique).

Dans ce contexte, le produit mécatronique impose de fait l'interpénétration entre les supports (fort couplage technologique entre les parties mécaniques, électroniques,... qui peuvent être physiquement fusionnées) (Figure 3-1).

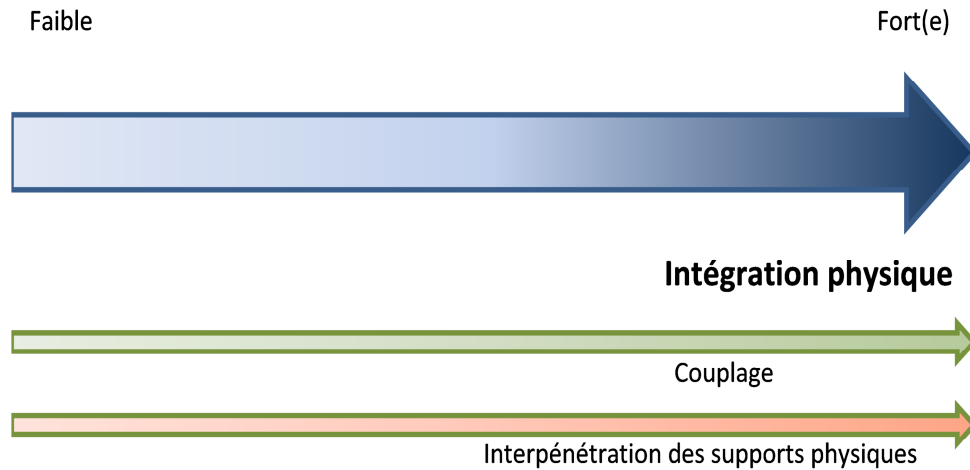


Figure 3-1 : Intégration physique d'un produit mécatronique

2.1.2 Une « r »évolution industrielle rapide

Dès lors, cette haute intégration fonctionnelle, multi-domaine et physique et les couplages correspondants rendent la conception des systèmes mécatroniques particulièrement complexe (Isermann 2007). Cette complexité croît avec l'augmentation du nombre de composants et de disciplines à intégrer dans un volume toujours plus compact (Craig 2009). Pourtant, la mécatronique a permis d'augmenter la compacité des objets courants de notre quotidien, qui sont géométriquement de plus en plus intégrés et légers comme l'iPod (Apple), l'alternateur i-Stars (Valeo) ou encore les serrures ou clés électroniques (Figure 3-2).



Figure 3-2 : Exemples de systèmes mécatroniques très intégrés physiquement.

Globalement, cette réduction du volume a également permis de réduire la quantité de matériaux et la masse du système.

L'exemple du moteur-roue (Figure 3-3), dont le développement a permis la suppression de la transmission mécanique, une diminution du poids et de l'encombrement, illustre bien cette tendance (Espanet 1999).

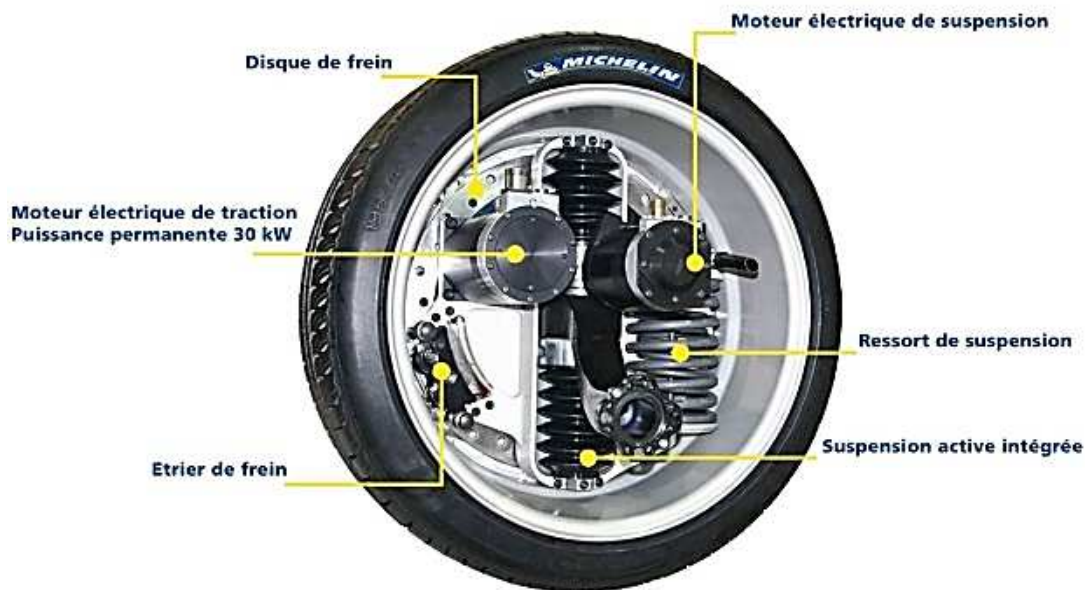


Figure 3-3 : Active Wheel de Michelin⁴⁰

L'évolution de cette intégration au cours du temps est, pour les systèmes mécatroniques, de plus en plus explicite. En effet, il y a quelques années, l'électronique était séparée de la mécanique et placée à côté alors que désormais l'électronique et la mécanique (support de la mécatronique) sont intégrées, allant même jusqu'à être fusionnées (Figure 3-4).

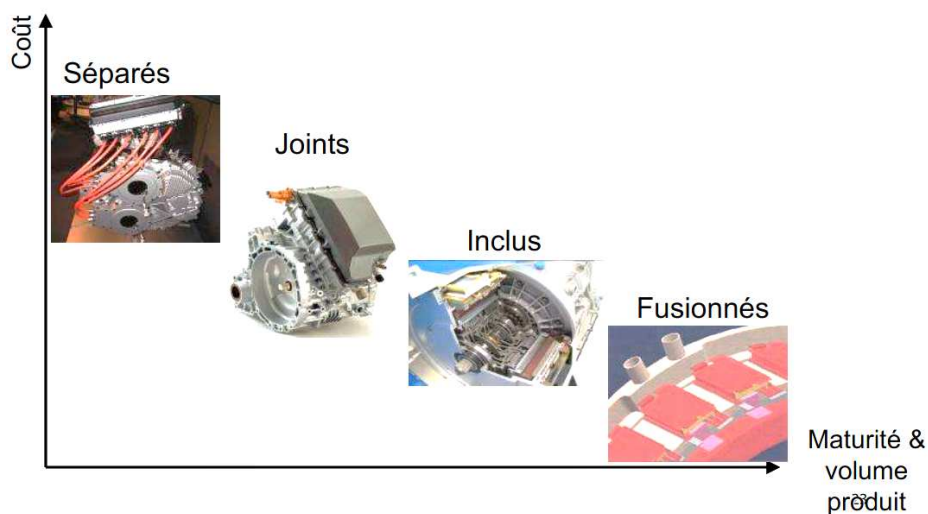


Figure 3-4 : Une haute intégration du système (source : Projet O2M³)

2.1.3 De nouveaux moyens et méthodes de conception et de fabrication

Aujourd'hui, l'électronique est insérée dans la mécanique et devient donc 3D. Cette nouveauté fait fortement évoluer le métier de la conception électronique et ses modèles/outils 2D vers de la conception mécatronique volumique (2,5D / 3D) avec pour l'instant très peu d'outils de modélisation adaptés⁴¹ (Krebs and Šegmanović 2013 ; Roumizadeh et al. 2009a). Pourtant, il existe déjà de nouveaux supports de circuit électronique 3D appelé MID (Molded Interconnected Devices) (Figure 3-5) ou encore FLEX

⁴⁰ <http://www.autoevolution.com/news/michelin-will-make-the-active-wheel-available-in-2010-2446.html>

⁴¹ <http://www.zuken.com/fr>

qui permettent de concevoir des systèmes électroniques/électriques sur des surfaces en 3D (Moser and Krause 2006; Roumizadeh et al. 2009c).

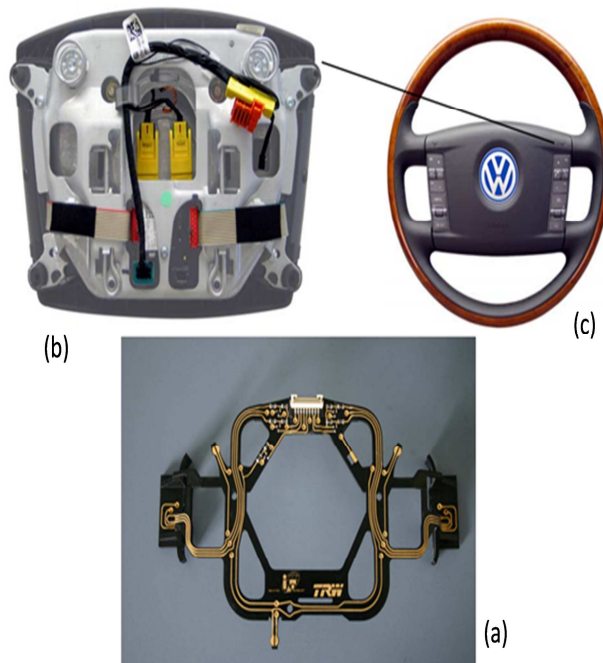


Figure 3-5 : Exemple d'applications MID remplaçant le câblage conventionnel d'un volant de direction multifonction d'une voiture (Heininger et al. 2014)

Le projet O2M (Outils de Modélisation Mécatronique)⁴² et notamment le sous-projet SP6 « Elec3D » (conception électronique 3D intégrée), s'est penché sur ce problème. Le but était, entre autres, de supprimer les ruptures entre les outils de modélisation et simulation mécaniques et électroniques, en offrant un routage 3D avec l'intégration des caractéristiques électroniques et électriques, dans un environnement de CAO mécanique 3D (Figure 3-6), tout en simplifiant la géométrie importée pour coupler les éléments finis et la simulation 2D électrique/électronique et enrichir ainsi la simulation 2D dans la perspective de gagner en fiabilité et rapidité.

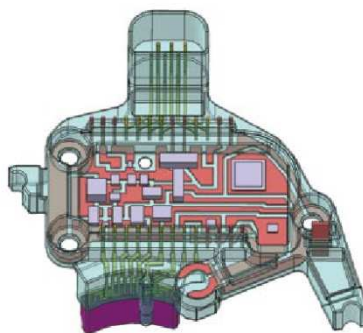


Figure 3-6 : Module de puissance 3D de chez Valeo⁴²

Par ailleurs, la problématique d'intégration physique entre les composants mécaniques et électroniques oblige parfois les concepteurs à penser autrement et à modifier la forme conventionnelle des composants. Citons l'exemple de la société Daimler Chrysler, qui a

⁴² Projet O2M (Outils de Modélisation Mécatronique) labellisé par Mov'eo et System@tic, financé par le FUI (Fond Unique Interministériel)

intégré un convertisseur de puissance et sa partie commande dans le moteur électrique d'un Groupe MotoPropulseur Electrique (GMPE) (Figure 3-7), pour disposer d'un GMPE mécatronique compact (Hammadi 2012). Le module de conversion de puissance a évolué vers une nouvelle géométrie à forme circulaire pour pouvoir être intégré au stator. Quant aux modules de puissance, ces derniers ont été modifiés sur la base d'une forme courbe afin de respecter l'enveloppe du montage. Cette migration montre la nécessité de nouvelles méthodes et techniques de fabrication et d'assemblage.

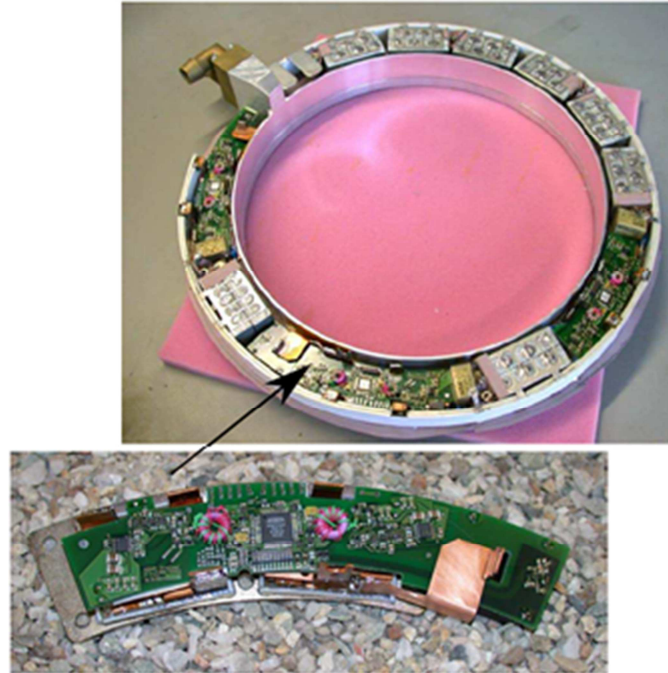


Figure 3-7 : Module de puissance du convertisseur d'un GMPE (Hammadi 2012)

Nous pouvons aussi citer la technologie 3D-MID (Roumizadeh et al. 2009c) et la technologie Flex évoquées précédemment qui permettent d'intégrer des fonctions mécaniques et électroniques en un seul élément de taille minimale. Les dispositifs MID peuvent être ainsi définis comme une pièce en plastique moulée par injection intégrant des fonctionnalités à la fois électriques et mécaniques dans un seul composant. Le circuit électronique est intégré dans le boîtier, ce qui permet d'augmenter significativement la structure compacte et la compacité fonctionnelle des composants (Islam et al. 2009). Par rapport aux circuits conventionnels, la technologie MID offre une plus grande liberté de conception. Ces technologies sont aujourd'hui utilisées dans de nombreuses applications de conception de produits mécatroniques comme les automobiles, les avions, les ordinateurs, les téléphones portables, les systèmes de contrôle industriels et médicaux, les satellites...

Enfin, il est intéressant d'évoquer l'évolution technologique opérée par la société Molex⁴³. Celle-ci a développé un processus combinant la technologie MID avec le processus Laser Direct Structuring (LDS) pour permettre l'intégration de fonctions mécaniques et électriques complexes dans des applications médicales compactes (Dixon 2012). En effet, le processus Laser Direct Structuring (LDS) permet de produire avec une haute résolution des circuits tracés sur des structures porteuses complexes en trois dimensions, intégrant ainsi les boîtiers et les PCB, qui étaient auparavant séparés, dans une seule unité (Heininger et al. 2014).

⁴³ <http://www.molex.com/molex/index.jsp>

2.1.4 Conséquences sur la conception mécatronique

Cette proximité des composants amène certes une plus grande compacité mais peut également engendrer des couplages multi-physiques, comme en attestent les travaux d'Ooshima (Ooshima and Masukata 2013) sur les interactions multi-physiques liées à la miniaturisation d'un système, ainsi que les nombreuses études sur les couplages multi-physiques des systèmes mécatroniques (Samin et al. 2007; Choley et al. 2010; Da Silva et al. 2012; Thouviot 2013).

Or même si ceux-ci sont généralement préjudiciables pour le fonctionnement du système et qu'il est souvent nécessaire de trouver les moyens de les éviter, ou tout du moins, de les minimiser (Roumizadeh et al. 2009b), certains peuvent parfois aussi conduire à des fonctions supplémentaires bénéfiques pour le système. Par exemple, dans le cas du moteur-roue (Figure 3-3), le fait que le moteur soit intégré dans la roue permet de réaliser une fonction supplémentaire qui est de protéger le moteur des chocs extérieurs (Espanet 1999). Prenons un autre exemple industriel : dans le roulement à billes instrumenté de SKF (Message and Landrieve 2002), le contrôle du confinement de l'entrefer entre la cible et le capteur permet la limitation de la sensibilité du capteur aux champs magnétiques externes (Figure 3-8).

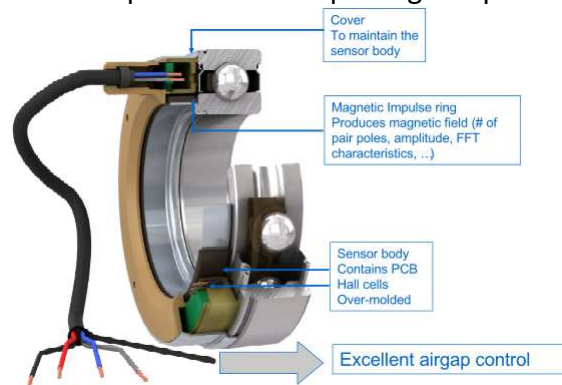


Figure 3-8 : Illustration des avantages de l'intégration physique sur un système mécatronique : le roulement à billes instrumenté (SKF™)

Toutefois, cette intégration physique se traduisant par des systèmes mécatroniques plus compacts, n'est pas sans conséquence sur d'autres aspects, tels que la maintenabilité et l'assemblage (voir chapitre 1 §4.3), qui deviennent quant à eux plus difficiles (concernant notamment l'accessibilité des composants, le remplacement de composants défectueux ou usagés, la gestion de configuration avec des modules et / ou composants interchangeables, etc.). Dans ce contexte, le concepteur doit alors trouver un compromis entre compacité et intégration des composants ainsi qu'entre accessibilité, maintenabilité et modularité des composants.

Quoiqu'il en soit, ce défi impacte directement le placement des composants. Si habituellement, le placement des composants n'intervient qu'à la fin du cycle de conception, c'est-à-dire dans la phase de conception détaillée, les considérations précédentes montrent qu'il est important de le prendre en compte dès que possible dans le cycle de conception (Gupta and Okudan 2008). En effet, la géométrie et le positionnement des composants deviennent fondamentaux dès la préconception pour traiter l'intégration physique des systèmes mécatroniques complexes (Plateaux et al. 2009; Warniez et al. 2014b; Barbedienne et al. 2014).

2.2 Modélisation géométrique et placement des composants

Pour pouvoir répondre au problème de placement des composants, il faut préalablement modéliser leur forme géométrique. Dans ce paragraphe, nous faisons donc l'état de l'art à la fois de la modélisation géométrique des objets et de leur placement. En ce qui concerne les métriques d'intégration physique développées dans ce paragraphe, elles adressent la phase amont de préconception (et non celle de conception détaillée), aussi nous prenons comme hypothèse que la géométrie considérée sera relativement simplifiée.

2.2.1 Modélisation géométrique

De nombreuses techniques ont été développées depuis longtemps, pour la représentation géométrique des objets (Pouget 1992) et notamment pour répondre au besoin croissant de la modélisation des formes complexes, appuyées par le fort développement d'outils CAO depuis les années 80 (Gillet 1994; Gardan 2002).

Ainsi pour la géométrie simplifiée, deux modèles principaux se distinguent :

- La méthode « *Semantic Feature Model* » (SFM) (Bidarra and Bronsvoort 2000) est adaptée pour créer une géométrie de base, définie au travers de paramètres de forme. La géométrie du système est construite en utilisant des opérations (addition et soustraction) sur d'autres composants géométriques. Cette modélisation a été enrichie par le modèle « *Declarative Family of Object Model* » (DFOM) (van der Meiden and Bronsvoort 2007). Ce dernier est basé sur la théorie des catégories définissant ainsi des familles de composants, et utilise l'approche algébrique. Ainsi, en plus des opérations de la méthode SFM, il comprend celles de la théorie des catégories comme l'union et l'intersection (Barbedienne et al. 2014).
- La deuxième approche, issue des modèles de tolérancement, est celle de la théorie « *TTRS* », « *Technologically and Topologically Related Surfaces* » (en français notée SATT «Surfaces Associées Technologiquement et Topologiquement») a été développée par Clément et al. (Clément et al. 1998). Cette théorie a été créée en 1986 pour générer la modélisation des objets en utilisant une approche déclarative. Ce modèle prend en compte les deux aspects -technologique et topologique- de deux surfaces en contact. La théorie des SATT est présentée comme une structure unifiée pour la représentation des objets et des contraintes géométriques pour le positionnement relatif des composants (Serré 2000). Une SATT est créée en cherchant les déplacements laissant invariant l'ensemble constitué de l'union de deux surfaces, définissant ainsi sept classes de surfaces selon leur degré respectif d'invariance cinématique (Table 3-1). Chaque classe est alors décrite par un Elément Géométrique de Référence Minimum (EGRM ou MRGE en anglais) qui permet le positionnement aisé de l'objet dans un espace euclidien. L'EGRM est une combinaison des éléments de situation suivants : un point, une ligne, une droite.

Classes SATT	Complexe	Prismatique	Révolution	Hélicoïdale	Cylindrique	Plane	Sphérique
Degrés d'invariance	0 (identité)	1 translation	1 rotation	1 rotation & 1 translation combinées	1 translation & 1 rotation	2 translations & 1 rotation	3 rotations
EGRM	Point		Point				Point
	Droite	Droite	Droite	Hélice	Droite		
	Plan	Plan				Plan	

Table 3-1 : Les 7 classes de surface et leurs EGRM (Clément et al. 1998)

Cette théorie a été adoptée par les normes internationales ISO 17450 (ISO 2011b; ISO 2012) et implémentée dans le logiciel de CAO Catia de Dassault Systèmes⁴⁴, pour faciliter la gestion des contraintes de tolérancement, de positionnement et d'assemblage, en simplifiant la représentation géométrique des éléments.

2.2.2 Positionnement des objets

En CAO, il existe plusieurs approches pour classer les modèles de représentation et de positionnement des objets géométriques, dont les plus répandus sont les suivants (Moinet et al. 2007) :

- Le modèle B-REP (Boundary Representation) modélise la frontière entre la peau de l'objet et le côté où se trouve la matière. Il contient deux types d'information : l'information géométrique (qui permet de positionner les objets dans l'espace notamment par les sommets pour les polyèdres) et l'information topologique (qui permet de relier les composants entre eux).
- Le modèle CSG (« Constructive Solid Geometry » ou arbre de construction) conserve un historique des séquences de construction. Il est constitué d'objets et des opérations (principalement booléennes) qui leur ont été appliquées.

Concernant les techniques de modélisation du positionnement relatif des composants, elles sont beaucoup plus nombreuses que celles pour la modélisation géométrique, car le challenge est beaucoup plus complexe. Par exemple, dans (Roussel 1994), l'auteur utilise la représentation des objets en 2D avec le codage par peigne de contour, pour optimiser le placement d'objets quelconques pour la découpe de plaques.

Concernant le positionnement relatif d'éléments géométriques moins complexes, la méthode d'analyse du degré de liberté (Degree Of Freedom Analysis) (Kramer 1991; Bettig and Shah 2001; Kim et al. 2006; Bettig and Kale 2012) est très utilisée, car elle est très facile à modéliser. Cette méthode est basée sur un marqueur (noté m), qui comprend un point, et deux axes orthogonaux x et z. Tous les éléments sont composés de marqueurs, et il est possible de leur appliquer 7 contraintes pour placer les composants.

La modélisation géométrique basée sur les SATT présentée dans le paragraphe précédent gère également la composition d'autres éléments SATT et leur positionnement relatif, en proposant 13 contraintes géométriques entre les EGRM (Table 3-2) pour les positionner et les orienter (Clément et al. 1998). Ces contraintes définissent les coïncidences, les distances, les angles entre les éléments de deux EGRM.

⁴⁴ <http://www.3ds.com/fr/produits-et-services/catia>

Reclassing case of MRGE and induced constraints	Line (cylindric) (C_C)	Plan (Plan) (C_P)	Point (Spherical) (C_S)
Line (cylindric) (C_C)	$D1=D2 (C_C) : C11$ $D1//D2 \text{ \& } D1 \neq D2 (C_T) : C12$ else (C_X) : C13	$D2 \perp P1 (C_R) : C8$ $D2 // P1 (C_T) : C9$ Else (C_X) : C10	$O1 \in D2 (C_R) : C4$ Else (C_X) : C5
Plan (Plan) (C_P)		$P1//P2 (C_P) : C6$ else (C_T) : C7	$(C_R) : C3$
Point (Spherical) (C_S)			$O1 = O2 (C_S) : C11$ Else (C_R) : C2

Table 3-2 : Les 13 classes de contraintes entre SATT (Clément et al. 1998; Choley 2005)

2.3 Evaluation des architectures spatiales

2.3.1 En milieu industriel

Aujourd'hui dans de nombreuses entreprises, un expert géomètre est en charge de définir, à partir de l'architecture physique fournie par l'architecte système, les volumes alloués à chaque entité, que ce soit par domaine (mécanique, capteurs, électronique...) ou par sous-système technologique. Cette allocation de volume résulte généralement de leur expérience sur les générations précédentes de produits ou de lois empiriques. Le placement des composants se fait alors suivant des critères propres à chaque domaine, souvent par des règles de ségrégation, préalablement établies par type de composants (exemple, les chapitres ATA dans l'aéronautique) ou par expérience. Ainsi l'évaluation de l'architecture spatiale se réduit au respect ou non du volume d'encombrement alloué à chacune des équipes et à la vérification en conception détaillée par des calculs en éléments finis, longs et coûteux, que les couplages multi-physiques engendrés ne nuisent pas aux performances du système.

Pour les systèmes mécatroniques, qui requièrent la collaboration de plusieurs équipes multi-domaines, la validation de l'architecture spatiale résulte donc généralement d'un processus de compromis difficile.

2.3.2 Approche mathématique

Pour évaluer la proximité des composants, l'approche mathématique la plus évidente est de mesurer la *distance* entre les composants. Voici donc quelques définitions de distances. Dans ce paragraphe, les termes mathématiques notés avec un astérisque « * » sont définis en Annexe 5.

2.3.2.1 Propriété d'une distance

La distance sur un ensemble E est une application d de $E \times E$ dans l'ensemble des \mathbb{R}^+ des nombres réels positifs ou nuls telle que, quels que soient les éléments x, y , et z de E , nous pouvons vérifier les propriétés suivantes :

La symétrie : $\forall x, y \in E, d(x, y) = d(y, x)$

La séparation : $\forall x, y \in E, d(x, y) = 0 \Leftrightarrow x = y$

L'inégalité triangulaire : $\forall x, y, z \in E, d(x, z) \leq d(x, y) + d(y, z)$

Il existe différentes distances comme la distance entre deux points (distance de Manhattan, distance euclidienne), ou entre un point et un ensemble, ou deux ensembles. Tout au long de ce paragraphe, pour définir ces distances, nous prenons dans \mathbb{R}^n un point dans chacun de ces ensembles : $(x_1, x_2, \dots, x_i, \dots, x_n)$ et $(y_1, y_2, \dots, y_i, \dots, y_n)$.

- Distance de Manhattan (\mathbb{R}^1)

La distance de Manhattan est la distance entre deux points dans une grille (Figure 3-9) basée sur un chemin strictement horizontal et /ou vertical (c'est à dire selon les lignes de la grille), par opposition à la diagonale. La distance de Manhattan est simplement la somme des composantes horizontales et verticales. Elle est définie par :

$$\sum_{i=1}^n |x_i - y_i| \quad (3.1)$$

Illustration :

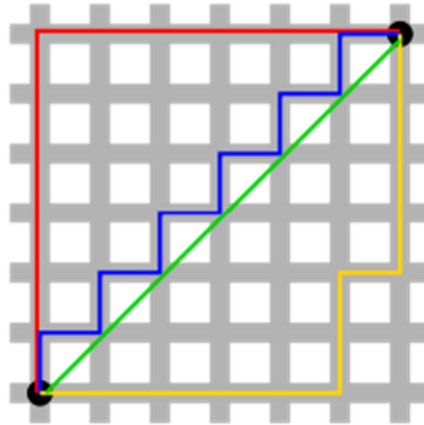


Figure 3-9: Distances de Manhattan (chemins rouge, jaune et bleu) et Distance euclidienne (en vert)

- Distance euclidienne (\mathbb{R}^2)

La distance euclidienne permet de généraliser l'application du théorème de Pythagore à un espace de dimension n . Elle est définie par :

$$\sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.2)$$

La distance euclidienne est la diagonale représentée en vert sur la Figure 3-9.

- Distance de Minkowski ($\mathbb{R}^{n=p}$)

La distance de Minkowski peut être considérée comme la généralisation de la distance de Manhattan et de la distance euclidienne. Elle est souvent utilisée pour les cas où $p = 1, 2$ ou ∞ .

Pour tout nombre réel positif $p > 0$, nous avons :

$$\sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p} = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \quad (3.3)$$

- Distance entre deux ensembles

Soient A et B deux ensembles dans \mathbb{R}^n muni d'une distance d définie par :

$$d(A, B) = \inf\{d(a, b) / (a, b) \in A \times B\} \quad (3.4)$$

- Distance d'un point à un ensemble

La distance d'un point à un ensemble est un cas particulier de la définition précédente (3.4) lorsqu'un des deux ensembles est réduit à un point.

Si A est un ensemble non-vide et x un élément de l'espace métrique E , la distance devient d de x à A dévient :

$$d(x, A) = \inf\{d(x, a), a \in A\} \quad (3.5)$$

C'est le rayon de la plus grande boule ouverte de centre x qui ne rencontre pas A .

Les deux distances précédentes relatives aux ensembles ne vérifient pas les propriétés des distances définies précédemment. Notamment la propriété de séparation : lorsque la distance est nulle, nous ne pouvons pas en déduire que ces deux ensembles sont égaux. Cependant, il est possible de définir une distance entre deux ensembles respectant les trois propriétés. Cette distance est la distance d'Hausdorff que nous détaillons dans le paragraphe suivant.

2.3.2.2 Définition de la distance d'Hausdorff

La distance d'Hausdorff est un outil topologique qui mesure l'éloignement de deux sous-ensembles d'un espace métrique sous-jacent.

Intuitivement, la distance d'Hausdorff permet de définir la distance entre deux ensembles. Prenons l'exemple d'un carré et d'un disque (Figure 3-10)

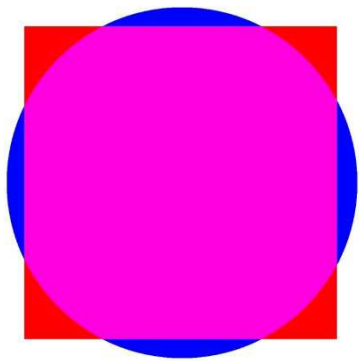


Figure 3-10 : Exemple de deux ensembles : un carré et un disque

La différence entre les deux solides est représentée par 4 lunules bleus et 4 presque triangles rouges. Le point du carré le plus éloigné par rapport au disque est le sommet du carré, à une distance a du disque. Le point le plus éloigné du disque par rapport au carré est le sommet de la lunule, à une distance b du carré (Figure 3-11). La distance d'Hausdorff est la plus grande valeur des deux, c'est-à-dire dans notre exemple a .

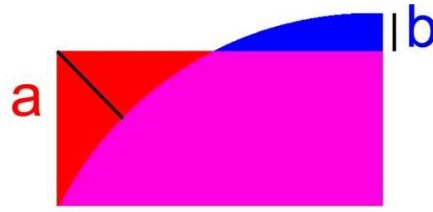


Figure 3-11 : Représentation des distances d'Hausdorff a et b

Soit X et Y , deux ensembles fermés bornés non vides d'un espace métrique* (E, d) , alors la distance d'Hausdorff est définie par l'expression suivante :

$$H(X, Y) = \max\{h(X, Y), h(Y, X)\} \quad (3.6)$$

Où h est la demi-distance d'Hausdorff dirigée, donnée par :

$$h(X, Y) = \sup_{x \in X} \inf_{y \in Y} d(x, y) \quad (3.7)$$

Cette distance $h(X, Y)$ trouve le point $x \in X$ dont la distance la plus proche d'un point y de Y est maximale parmi tous les points de X et donne la distance entre X et son point de plus proche de Y (Figure 3-12).

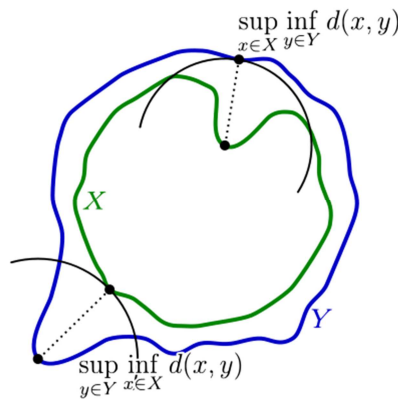


Figure 3-12 : Distance d'Hausdorff entre la ligne verte X et la ligne bleue Y⁴⁵

La distance d'Hausdorff est souvent utilisée dans le traitement d'images pour rechercher les similitudes entre les images (Vergeest et al. 2003; Yang 2008).

Une autre définition de la distance d'Hausdorff (Jaulin et al. 2001) s'exprime par :

$$h_{\infty}(A, B) \triangleq \max\{h_{\infty}^0(A, B), h_{\infty}^0(B, A)\} \quad (3.8)$$

Avec les demi-distances :

$$h_{\infty}^0(A, B) \triangleq \inf\{r \in \mathbb{R}^+ | A \subset B + rU\} \quad (3.9)$$

Avec A et B deux compacts* de \mathbb{R}^n et U la boule unité fermée*.

Pour obtenir $h_{\infty}^0(A, B)$, B doit croître jusqu'à ce qu'il contienne A et pour obtenir $h_{\infty}^0(B, A)$ A doit croître jusqu'à contenir B (Figure 3-13).

⁴⁵ http://en.wikipedia.org/wiki/Hausdorff_distance

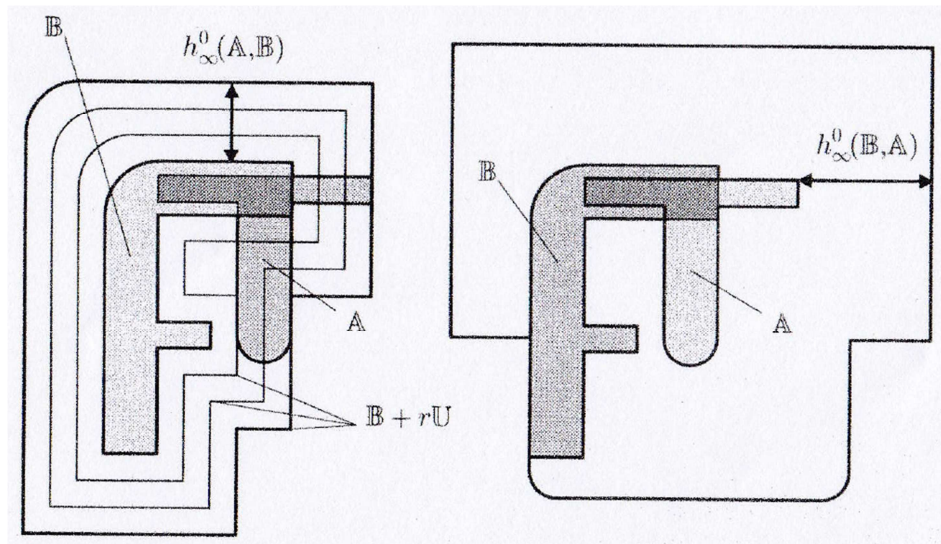


Figure 3-13 : Définition de la demi-distance d'Hausdorff (Jaulin et al. 2001)

2.3.3 Les méthodes et métriques existantes pour l'évaluation de l'intégration physique

En phase de préconception, l'architecte système valide les architectures fonctionnelles, logiques et physiques. L'architecture physique définit les composants du système. A ce stade de la conception, soit ces composants sont des composants sur étagère avec une forme, des dimensions géométriques et des caractéristiques physiques précises, soit un certain nombre de propriétés (géométriques telles que le volume d'encombrement ou la forme, dimensionnelles ou de performance) sont spécifiées pour le développement/la réalisation ultérieure du composant. Or aujourd'hui, très peu d'études se sont portées sur l'intégration de la géométrie dans la phase de préconception et notamment pour les spécifications géométriques dans le modèle « système ». Quelques auteurs ont cependant travaillé sur la prise en compte de la géométrie dans les phases amont de conception.

Baysal et al. (Baysal et al. 2005) proposent une méthode pour modéliser la géométrie et le positionnement des composants pour l'analyse du tolérancement en langage UML. Cependant, ce positionnement n'est pas relatif et n'intègre pas directement de contraintes. Il est alors difficile pour le concepteur de calculer le positionnement respectif de chaque partie. Par ailleurs, Moullec et al. (Moullec et al. 2013) proposent une approche intégrée basée sur les réseaux bayésiens et sur le problème de satisfaction de contraintes (CSP) afin d'évaluer les architectures candidates en intégrant l'optimisation du placement des composants. L'approche CSP (Constraint Satisfaction Problem) est proposée pour prendre en compte les contraintes du concepteur relatives aux exigences du système. La formalisation CSP consiste à définir : un ensemble de variables de décision, les domaines des variables de décision et un ensemble fini de contraintes qui doivent être satisfaites par les variables de décision. Les contraintes définies sont ici de 5 types : des contraintes de positionnement, des contraintes de proximité, des contraintes d'orientation, des contraintes de distance et des contraintes de performance. Le concepteur définit également le nombre de composants et leurs dimensions ainsi que celles de l'espace maximal disponible. Le problème CSP est alors résolu par un solveur et plusieurs architectures physiques sont proposées répondant aux exigences du système. Malheureusement, la géométrie des composants est, dans leur approche, limitée à des formes parallélépipédiques.

Enfin, d'autres études portent sur des métriques mesurant l'intégration d'un système, et notamment sur les conséquences de l'intégration physique. Ainsi, ces études proposent des métriques dites géométriques pour évaluer d'autres objectifs prenant en compte le

placement des composants. Le chapitre 1 §4.3 fait l'état de l'art de ces métriques comme celles pour la modularité des composants (Kremer et al. 2012), l'assemblage/ désassemblage (Guide et al. 1997; Das et al. 2000b), la recyclabilité (Scaravetti 2004; Atlee and Kirchain 2006), l'accessibilité (Das et al. 2000b; Hsu and Lin 2002), etc.

2.4 Problématique

Actuellement le problème concret des concepteurs est que l'architecte système valide les architectures fonctionnelles, logiques et physiques, mais gère rarement l'architecture spatiale. Celle-ci est souvent le résultat de simulations en conception préliminaire. Or en conception mécatronique, les équipes techniques multi-domaines ont souvent des méthodes, des objectifs et des règles de ségrégation qui leur sont propres. Par conséquent, les critères d'évaluation de leurs architectures spatiales sont relatifs à leur domaine, et s'appliquent difficilement aux systèmes mécatroniques dont l'intégration physique requiert de placer un maximum de composants dans un volume compact en limitant les couplages multi-physiques néfastes.

Ainsi, après avoir fait l'état de l'art sur l'intégration physique des systèmes mécatroniques et les méthodes pour l'évaluer, nous avons identifié un certain nombre de métriques basées sur le positionnement géométrique, sans toutefois avoir trouvé des métriques qui adressent directement l'intégration physique.

3 Métrique d'intégration physique développée

Comme soulevé au chapitre 1 §4, pour chaque phase du cycle de conception, des métriques peuvent être élaborées (**Erreur ! Source du renvoi introuvable.**) afin d'aider le concepteur à faire un choix.

Dans ce qui suit, nous nous intéressons aux métriques d'intégration physique dès la phase de préconception pour le choix d'architectures physiques. En effet, après avoir défini les différentes architectures physiques candidates qui allouent des composants physiques à des éléments logiques (ou organes) précédemment identifiés, le concepteur a besoin de critères et d'indicateurs en vue d'évaluer et de comparer ces architectures. Ces différents indicateurs permettront, à un plus haut niveau de considération, de construire une métrique d'intégration physique. L'objectif visé est de permettre à l'architecte système de prendre en compte les exigences géométriques pour faire ses choix d'architecture sans attendre la phase de conception détaillée.

La formalisation de ces exigences géométriques dans la phase de conception préliminaire, au travers d'une extension géométrique du langage SysML, permettra ainsi à l'ensemble des équipes techniques multidisciplinaires d'avoir une vue unique de ces spécifications en amont. Elles seront les données d'entrée pour leurs études dans leur domaine spécifique. Enfin, l'unicité de ces données facilitera les échanges et la démarche collaborative, notamment pour le choix de l'architecture finale.

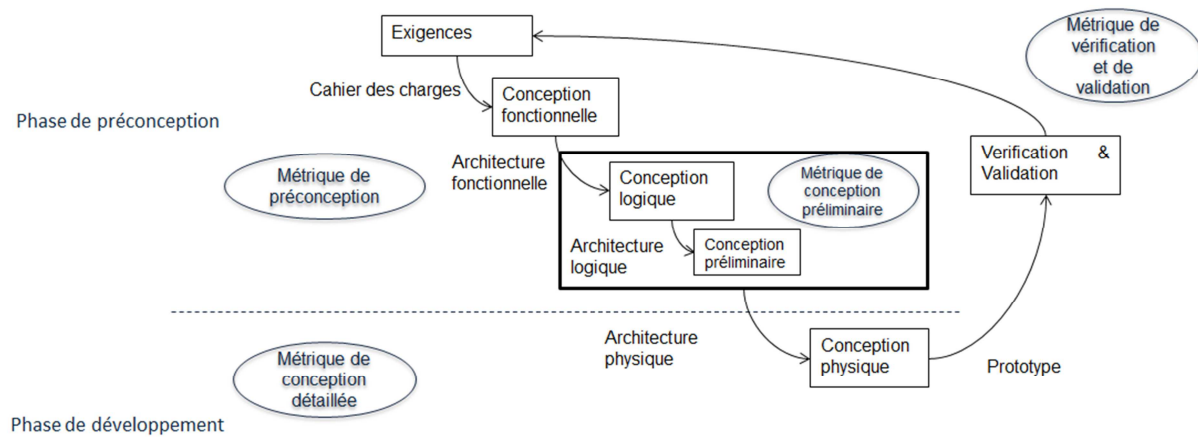


Figure 3-14 : Les métriques dans le cycle de conception

Cette métrique basée sur la géométrie et le positionnement des composants, doit permettre de mesurer l'intégration physique globale du système des différentes architectures candidates. Elle est basée principalement sur les données et contraintes géométriques telles que la forme des composants, leurs dimensions, leur volume, etc.

La caractéristique essentielle de la métrique globale d'intégration physique permettra de comparer plusieurs architectures physiques et spatiales. Une architecture physique est un ensemble de composants susceptibles de réaliser les fonctions élémentaires et de transporter les flux d'entrée/sortie de l'architecture logique retenue. Elle relie ces composants entre eux à l'aide de flux (géométriques, mécaniques, électriques, etc.). Ces composants sont également reliés aux composants des autres sous-systèmes puis aux éléments de l'environnement externe. Par conséquent, il existe de nombreuses architectures candidates différentes.

Dans ce chapitre, nous évaluons ces architectures candidates par rapport à leurs propriétés d'intégration et notamment par rapport à des indicateurs portant sur le volume disponible et l'accessibilité (**Erreur ! Source du renvoi introuvable.**).

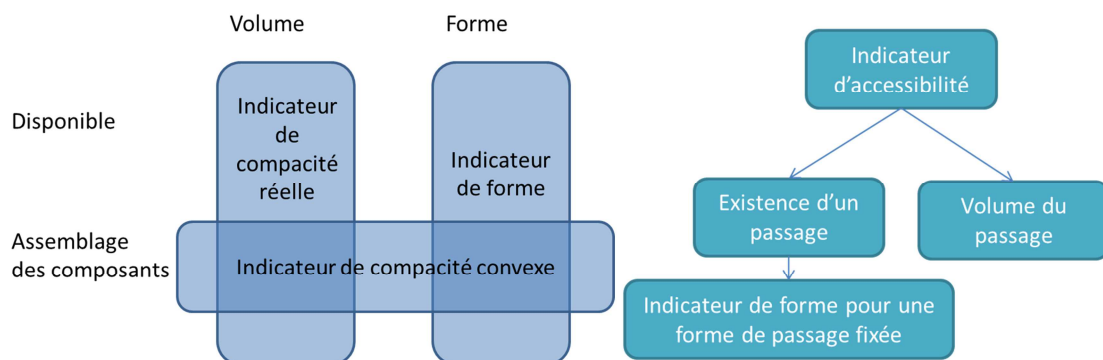


Figure 3-15 : Les différents indicateurs de la métrique d'intégration physique

Formulons comme hypothèses de départ, les éléments suivants :

- L'architecture physique donnée est constituée de composants réels ou à réaliser disposant d'un volume et d'une forme définis (sur étagère ou dont un certain nombre de données géométriques ont déjà été fixées pour le développement de ces composants).
- Les formes des composants ont des géométries simples (parallélépipède, sphère, cylindre, etc.)

- L'enveloppe disponible du système global a un volume et une forme qui peuvent être ou non connus (fixés).

3.1 Indicateurs de compacité

3.1.1 Enjeux et définitions

Par analogie à la définition de la compacité des matériaux en cristallographie qui se définit comme le rapport entre le volume occupé par des ions, atomes, molécules, etc. et le volume de la maille, la compacité d'un système sera définie comme le volume occupé par ses composants sur le volume disponible. Elle sera toujours inférieure à 1, une compacité de 1 signifierait qu'il n'y a aucun vide entre les éléments. L'enjeu de la compacité est donc de disposer les éléments, de sorte à en placer un maximum dans un volume minimum.

Sur la base de la définition précédente, l'objectif de ce premier indicateur est d'évaluer dans un premier temps si la totalité du volume d'encombrement disponible est occupé par le volume de l'assemblage des composants physiques ou au contraire s'il y a du vide à l'intérieur de ce volume. Dans ce cas de figure, le volume minimal d'occupation est la somme de tous les volumes des composants.

Pour cela, le concepteur compare plusieurs architectures physiques et spatiales constituées de composants physiques qui ont un volume et/ou une forme définis avec ou non une position d'assemblage donnée. Nous avons développé un indicateur de compacité qui se décompose en deux scénarios (**Erreur ! Source du renvoi introuvable.**) :

- Dans le cas où le volume d'encombrement est connu, l'indicateur de compacité réelle peut être calculé et notamment il permet d'identifier si le volume d'encombrement disponible est suffisant pour contenir de façon théorique l'ensemble des composants, indépendamment de leur agencement. Cet indicateur est une première information sur la possibilité ou non d'intégrer les composants dans le volume disponible.
De plus, si la forme de l'enveloppe d'encombrement et la topologie de l'assemblage sont connues, il est alors possible de savoir si l'architecture spatiale de l'assemblage des composants est intégrée dans cette forme, au travers d'un indicateur de forme. Cela donne une deuxième indication sur l'intégration physique des composants et de leur assemblage.
- Dans la situation où le volume d'encombrement disponible n'est pas forcément défini mais que l'assemblage est connu, l'enveloppe convexe de l'assemblage peut être prise en compte pour définir le volume disponible minimal et calculer l'indicateur de compacité « convexe » du système. Il définit alors la compacité de l'assemblage en considérant son volume convexe permettant d'évaluer le volume d'encombrement « idéal » (minimal) d'un tel assemblage. De même, si la forme du volume disponible est connue, nous pouvons calculer l'indicateur de forme, afin de connaître si l'enveloppe convexe de l'assemblage peut être contenue dans la forme du volume disponible.

3.1.2 Indicateur de compacité réelle

3.1.2.1 Expression

Ce premier indicateur s'évalue alors comme le rapport entre le volume réel de l'assemblage des composants et le volume d'encombrement disponible.

$$I_{CR} = \frac{\text{Volume réel de l'ensemble des composants}}{\text{Volume d'encombrement disponible}} = \frac{\sum V_i}{V_{ED}} \quad (3.10)$$

Avec I_{CR} : l'indicateur de compacité réelle

V_{ED} : le volume disponible

$\sum V_i$: la somme des volumes de chaque composant.

3.1.2.2 Bornes

Dans ce cas de figure, les bornes de cet indicateur ne sont plus normées.

- Lorsque la valeur de l'indicateur de compacité réel I_{CR} est comprise entre 0 et 1 cela signifie qu'il y a du vide et qu'il reste donc peut être de l'espace disponible. Plus la valeur se rapproche de 0 et plus il y a d'espace disponible.
- Lorsque la valeur d' I_{CR} est égal à 1 cela signifie qu'il n'y a aucun vide, la totalité du volume d'encombrement est occupée.
- Lorsque la valeur d' I_{CR} est supérieur à 1 cela signifie que le volume total des composants est supérieur au volume d'encombrement disponible. Il y a donc un problème d'intégration.

Ainsi, nous pouvons considérer que cet indicateur est valable lorsque sa valeur est comprise entre 0 et 1. En effet, si la valeur dépasse 1, l'architecture ne pourra pas être retenue, puisque les composants ne pourront pas être intégrés.

3.1.3 Indicateur de forme

3.1.3.1 Définition

Dans le cas où la valeur de l'indicateur de compacité réelle est inférieure à 1, que la forme du volume d'encombrement disponible et la topologie de l'assemblage sont également connues, il est alors possible de savoir si l'assemblage spatial des composants est contenu ou non dans la forme du volume d'encombrement disponible. En effet, l'indicateur de compacité réelle décrit précédemment ne prenait pas en compte la forme du volume alloué (par exemple la forme globale d'un carter) du système.

Pour illustrer l'intérêt de cet indicateur, prenons des cas simples.

Prenons un premier exemple d'une sphère et de 3 composants S_1 , S_2 et S_3 .

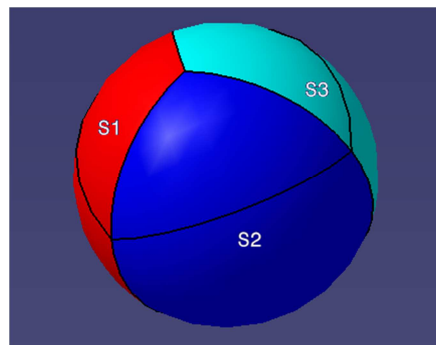


Figure 3-16 : Assemblage de 3 composants S_1 , S_2 et S_3

L'assemblage des 3 composants S_1 , S_2 et S_3 forme une sphère qui est parfaitement intégrée dans le volume d'encombrement disponible constituant également une sphère (Figure 3-16).

Le taux d'occupation réel est alors égal à 1 car il n'y a pas d'espace libre et qu'il n'y a pas non plus de problème d'intégration. L'indicateur de compacité convexe vaut également à 1 étant donné que les sphères sont des objets convexes. Aussi il n'y a pas de différences avec le taux de compacité réelle.

Prenons maintenant comme volume d'encombrement disponible, une forme cubique (le cube de couleur marron : Figure 3-17) dont le volume d'encombrement est égal au même volume que celui de la sphère de l'exemple précédent. Nous gardons les composants S_1 , S_2 et S_3 ainsi que leur assemblage.

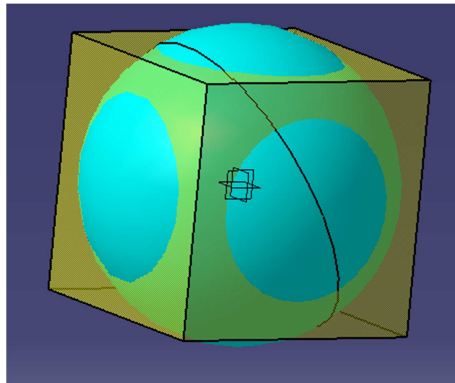


Figure 3-17 : Assemblage des composants S_1 , S_2 et S_3 dans un volume disponible cubique

Les indicateurs de compacité réelle et convexe sont, dans ce cas, encore égaux à 1 alors que les solides S_1 , S_2 et S_3 ne sont pas intégrés dans le volume d'encombrement disponible (le cube).

En effet, le volume de l'assemblage et le volume d'encombrement disponible n'ont pas changé : ils restent égaux dans les deux cas. Seule la forme du volume d'encombrement disponible a changé.

Ce nouvel indicateur de forme permet donc de prendre en compte la forme du volume disponible, ainsi que la topologie de l'assemblage des composants.

3.1.3.2 Expression

Nous cherchons à savoir si l'assemblage des composants est intégré dans la forme du volume d'encombrement disponible. Pour mesurer l'écart entre les deux formes, nous utilisons la demi-distance d'Hausdorff, équation (3.9) (décrite au §2.3.2.2 de ce chapitre) : $h_{\infty}^0(\mathbb{A}, \mathbb{B})$ où \mathbb{A} est l'ensemble du volume disponible et \mathbb{B} l'ensemble des composants assemblés. Le résultat de cet indicateur est un booléen :

- Si $h_{\infty}^0(\mathbb{A}, \mathbb{B}) > 0$, alors la forme du volume d'encombrement ne permet pas d'intégrer la topologie de l'assemblage dans cette forme. Le résultat de l'indicateur est faux.
- Lorsque $h_{\infty}^0(\mathbb{A}, \mathbb{B}) = 0$, cela signifie que la forme du volume disponible permet d'intégrer la topologie de l'assemblage des composants. Le résultat devient donc vrai.

3.1.4 Indicateur de compacité convexe

3.1.4.1 Expression

Cet indicateur permet de calculer la compacité d'un système, en considérant l'enveloppe convexe d'un assemblage donné comme son enveloppe disponible (minimale). Cet indicateur permet ainsi de comparer deux architectures même lorsque le volume

d'encombrement total n'est pas connu, par ailleurs, elle donne une information complémentaire sur l'intégration physique lorsque les deux premiers indicateurs sont déjà satisfaisants.

Définition d'ensemble convexe et d'enveloppe convexe (Klee and al. 1997) :

Un sous-ensemble C non vide de \mathbb{R}^n est dit convexe si pour tout couple x, y de points distincts de C , le segment $[x, y]$ est entièrement contenu dans C (Figure 3-18).

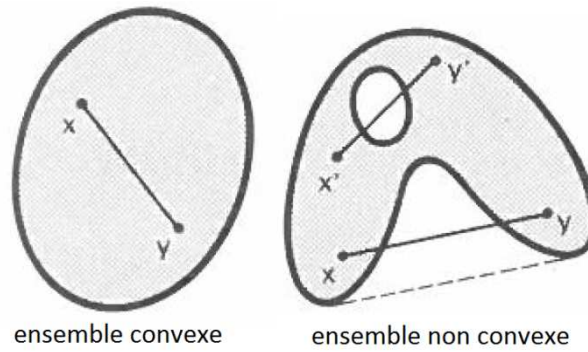


Figure 3-18 : Illustration d'ensemble convexe et non convexe

X est un sous-ensemble quelconque de \mathbb{R}^n , l'intersection \tilde{X} de tous les ensembles convexes contenant X est un ensemble convexe contenant X qui s'appelle l'enveloppe convexe de X (Figure 3-19). L'enveloppe convexe fermée de X est l'adhérence de \tilde{X} .

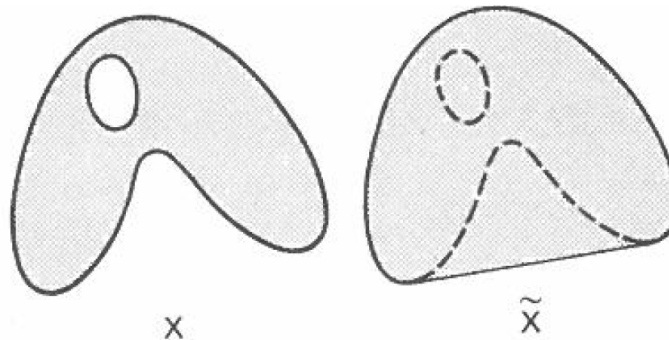


Figure 3-19 : Enveloppe convexe \tilde{X} d'un sous-ensemble X

L'indicateur de compacité convexe est le rapport entre le volume réel de l'assemblage des composants physiques et l'enveloppe convexe calculée de cet assemblage :

$$I_{CC} = \frac{\text{Volume réel de l'ensemble des composants}}{\text{Volume convexe}} = \frac{\sum V_i}{V_c} \quad (3.11)$$

Avec I_{CC} : l'indicateur de compacité convexe

$\sum V_i$: la somme des volumes de chaque composant

V_c : le volume de l'enveloppe convexe de l'assemblage

3.1.4.2 Bornes

Les bornes de cet indicateur sont comprises entre 0 et 1.

- Si la valeur d' I_{CC} (Indicateur de compacité convexe) est égale à 1, cela signifie qu'il n'y a aucun vide.

- Si la valeur d' I_{cc} se rapproche de 0, cela signifie qu'il y a du vide. Plus il se rapproche de 0 et plus il y a de l'espace disponible.

3.2 Indicateur d'accessibilité

3.2.1 Définition

L'indicateur d'accessibilité permet de connaître l'existence d'un passage et de son volume afin de disposer d'un accès au(x) composant(s) se trouvant à l'intérieur d'un autre et de pouvoir rajouter des composants dans l'espace disponible. L'accessibilité est un des critères de la maintenabilité (Chapitre 1 §4.3) et de la modularité (Chapitre 1 §4.2.1).

Lors d'une tâche de montage ou de maintenance, il s'agit de générer des trajectoires où les pièces considérées peuvent être montées ou extraites sans collision avec d'autres composants. Aussi, l'indicateur d'accessibilité permet d'évaluer s'il y a ou non un passage disponible pour accéder à un composant situé à l'intérieur d'un autre sans heurter les autres composants ou pour permettre l'insertion d'un nouveau composant dans l'espace disponible.

A ce titre, l'indicateur renseigne, par exemple s'il existe de la place pour passer un câble reliant un composant interne à l'extérieur. Il sera donc défini relativement à la taille des objets que l'on souhaite faire passer ou insérer.

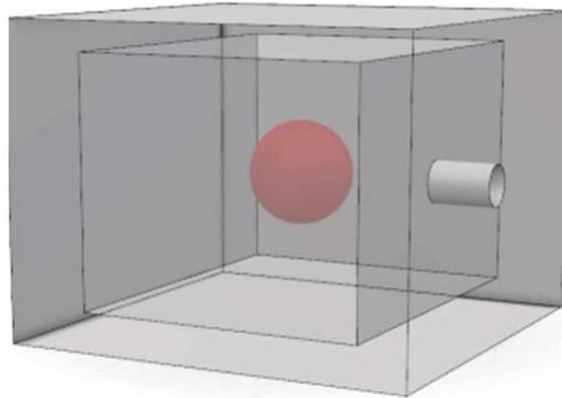


Figure 3-20 : Illustration d'un volume de passage

3.2.1.1 Expression

a) Existence d'un passage

Etant donnés (Figure 3-20)

- Un composant « enveloppe » $C_0 \subset \mathbb{R}^3$,
- Un point $x_0 \in \mathbb{R}^3 \setminus \text{Conv}(C_0)$, où *Conv* désigne l'enveloppe convexe,
- n composants $C_i \subset \mathbb{R}^3$, avec $1 \leq i \leq n$, deux à deux disjoints, à l'intérieur du composant « enveloppe » C_0 ,
- pour tout $1 \leq i \leq n$ un point $x_i \in C_i$,

Nous définissons un *chemin admissible*

$$f = f(x_0, x_i) \text{ de } x_0 \text{ à } x_i$$

comme un chemin qui ne passe par aucun autre composant à l'intérieur de C_0 que C_i , autrement dit comme une application continue de

$$f : [0,1] \rightarrow \mathbb{R}^3 \text{ telle que } f(0) = x_0, f(1) = x_i$$

et

pour tout $t \in [0,1]$ et tout $k \neq i, f(t) \notin C_k$.

b) Volume du passage

Soit $i \in \{1, \dots, n\}, x_i \in C_i$ et f un chemin admissible reliant x_0 à x_i .

Nous cherchons pour tout point sur ce chemin f , le plus grand rayon de la boule unitaire \mathbb{U} ne rencontrant aucun composant.

Pour tout $t \in I = [0,1]$, posons

$$r_t = \min_{k \neq i} d(f(t), C_k),$$

où $d(a, C)$ désigne la distance du point a à l'ensemble C . Ainsi, r_t représente la distance du point $f(t)$ situé sur le chemin considéré à un autre que C_i le plus proche de ce point.

Pour connaître, le plus petit passage possible, nous prenons le minimum de ces rayons.

Posons alors

$$r_i = \min_{t \in [0,1]} r_t$$

r_i est le plus grand rayon possible pour un tube centré sur le chemin f reliant x_0 à x_i et ne rencontrant pas les autres composants C_k avec $k \neq i$.

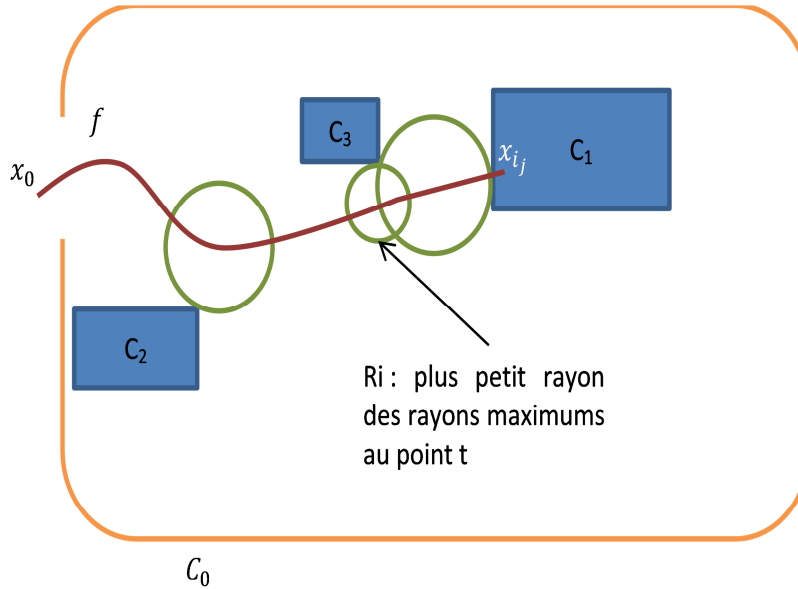


Figure 3-21 : Illustration en 2D du volume du passage disponible

L'indicateur d'accessibilité s'évalue alors comme suit :

$$I_a = \left(1 - \frac{r_{objet}}{r_i \text{ de tous les passages}}\right) \times \frac{\text{Nb de passages possibles}}{\text{Nb total de passages}} \quad (3.12)$$

Avec r_{objet} : rayon de la plus petite boule intégrant l'objet à faire passer

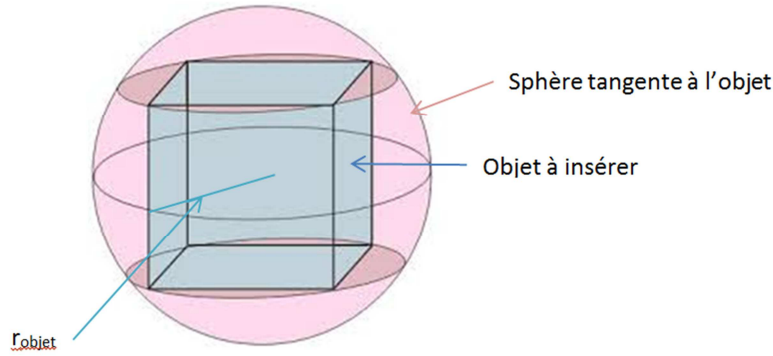


Figure 3-22 : Explication de r_{objet}

3.2.1.2 Bornes

L'indicateur a une valeur entre 0 et 1.

Lorsque la valeur de cet indicateur vaut 0, cela signifie qu'aucun volume de passage ne permet d'acheminer un objet vers le composant souhaité et donc que le rayon minimum des rayons maximum au point y est plus petit que le rayon minimum de la boule intégrant l'objet à faire passer.

Lorsque la valeur de l'indicateur se rapproche de 1, cela signifie que le minimum de tous les rayons maximum des chemins possibles est bien plus grand que le rayon minimum de la boule intégrant l'objet à faire passer et que le nombre de passages possibles est égal au nombre total de passages .

3.2.1.3 Exemples

Pour expliquer cet indicateur, prenons par exemple un objet dont le rayon de la plus petite boule l'intégrant est de 4 centimètres et le rayon minimum de passage est de 2,5 centimètres. Dans ce cas, il peut y avoir n passages, l'indicateur aura comme valeur :

$$I_a = \left(1 - \frac{4}{2,5}\right) \times \frac{0}{n} = 0$$

Prenons un autre cas, r_{objet} vaut 1,5 centimètres et r_i est égale à 5 centimètres. Tous les passages sont des passages possibles, l'indicateur vaut alors :

$$I_a = \left(1 - \frac{1,5}{5}\right) \times \frac{n}{n} = 0,7$$

Pour finir, r_{objet} vaut toujours 1,5 centimètres ainsi que r_i reste égale à 5 centimètres. La moitié des passages sont des passages possibles. L'indicateur prend la valeur suivante :

$$I_a = \left(1 - \frac{1,5}{5}\right) \times \frac{n/2}{n} = 0,35$$

3.3 Métrique d'intégration physique (MIP)

3.3.1 Expression

La métrique d'intégration physique est composée des deux indicateurs décrits précédemment : un indicateur de compacité (réelle (Eq. (3.10)) ou convexe (Eq. (3.11)) suivant les données disponibles) et de l'indicateur d'accessibilité (Eq. (3.12)) :

$$MIP_1 = \lambda_1 I_{cc} + \lambda_2 I_a \quad (3.13)$$

$$MIP_2 = \lambda_1 I_{cr} + \lambda_2 I_a \quad (3.14)$$

Comme dans le Chapitre 2, nous avons choisi d'utiliser une approche multicritères (puisque nous avons ici plusieurs indicateurs) basée sur une somme pondérée. Rappelons que les coefficients λ_i sont choisis par les différents concepteurs avant la phase de préconception pour pondérer chaque indicateur selon ses objectifs de conception. Ces coefficients ont une valeur comprise entre 0 et 1 et leur somme doit être égale à 1.

Selon les deux cas définis au §3.1.1, il existe donc deux métriques d'intégration physique.

3.3.2 Discussion

La métrique d'intégration physique est composée de deux indicateurs dont un indicateur de compacité. Or généralement, dans la conception des systèmes mécatroniques, l'objectif n'est pas seulement de réduire le volume mais également la masse. Il serait donc intéressant par la suite de développer un indicateur de densité, prenant en compte en plus la masse.

Par ailleurs, la compacité, au sens où nous l'avons définie n'est pas toujours une fonction qu'il faut à tout prix maximiser. En effet, certains espaces vides peuvent avoir une fonctionnalité. A titre d'exemple, citons les disjoncteurs disposant d'un espace vide libre pour permettre d'assurer la fonction d'isolation électrique. Parfois, le vide peut aussi servir pour l'isolation thermique ou la cinématique d'un système mobile. L'idée est donc dans ce cas-là d'ajouter à l'architecture physique des « composants fictifs » de volume de vide utile, pour ne pas altérer la pertinence des indicateurs précédemment développés.

Enfin, dans le milieu industriel, des règles de placement de composants existent, notamment les règles de ségrégation pour éviter les risques de couplages multi-physiques. Ainsi, les experts du bureau d'études savent qu'un composant A ne doit pas se trouver à moins d'une distance "x" d'un autre composant B et ce pour des raisons de couplages thermiques ou de compatibilité électromagnétique (CEM). Malheureusement ces règles sont rarement connues des architectes systèmes, qui ne peuvent ainsi que « pré-valider » des architectures spatiales, sans présager des couplages multi-physiques potentiels. Un travail collaboratif avec Romain Barbedienne, en thèse sur le développement d'un outil de positionnement de composants en préconception sous contraintes thermiques, permettra de développer de nouvelles métriques de couplages multi-physiques pour réduire cette limite actuelle.

4 Exemple mécatronique

En remarques préliminaires, comme nos travaux portent sur la phase de préconception, les composants seront représentés par des formes simples (parallélépipède rectangle et cylindre). Les architectures ont été modélisées avec le logiciel Catia V5. Cependant, comme nous travaillons en préconception et donc n'avons besoin de positionner que des volumes simplifiés, ces architectures auraient pu être modélisées avec un outil CAO moins complexe que le logiciel Catia. Un outil 3D plus adapté pour la phase de préconception est actuellement en cours de développement dans le cadre des travaux de thèse de Romain Barbedienne.

De plus, la modélisation par un outil CAO avait été initialement choisie pour calculer les indicateurs de compacité convexe et d'accessibilité, qui nécessitent de connaître la topologie de l'assemblage. Malheureusement, faute de temps, nous n'avons pas pu implémenter les

algorithmes de calcul de ces indicateurs dans le logiciel CAO, notre expérimentation se limitant donc à l'indicateur de compacité réelle.

4.1 Présentation du système et des différentes architectures

Le système mécatronique industriel choisi est le système hybride de la Toyota Prius.

Un système hybride est la combinaison d'un moteur à combustion et d'un moteur électrique, pour tirer parti des avantages de chaque source d'énergie tout en compensant les lacunes de l'autre.

Il existe 3 types de systèmes hybrides avec chacun des architectures physiques et spatiales différentes (Toyota Motor Corporation 2003).

4.1.1 Le système hybride série

Le moteur thermique (Engine) entraîne une génératrice (alternateur) (Generator) qui produit de l'électricité. Le moteur électrique (Motor) utilise l'électricité générée par la génératrice pour entraîner les roues (Figure 3-23). Par conséquent, le moteur thermique et le moteur électrique fournissent la même quantité de travail. - Le moteur thermique ne sert qu'à recharger les batteries. Le moteur électrique permet d'animer les roues. Cette configuration génère et fournit de l'électricité au moteur électrique et charge efficacement la batterie (Battery). Il y a donc deux moteurs : une génératrice (qui a la même structure qu'un moteur électrique) et un moteur électrique.

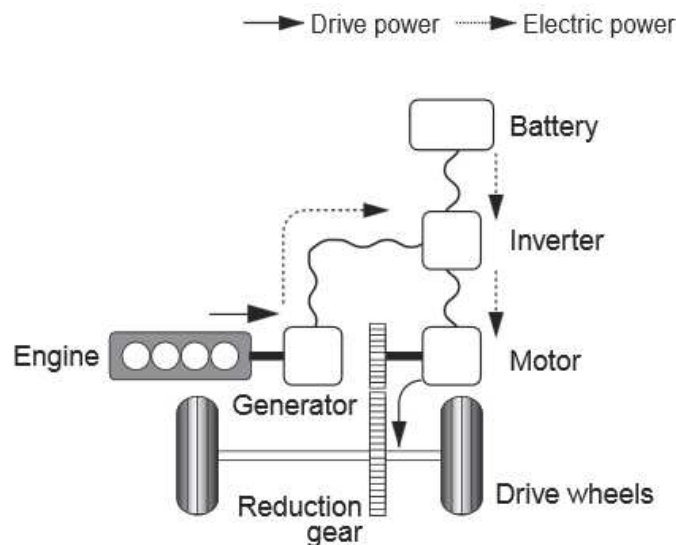


Figure 3-23 : Système hybride série (Toyota Motor Corporation 2003)

4.1.2 Le système hybride parallèle

Dans le système hybride parallèle, le moteur thermique et le moteur électrique entraînent tous les deux les roues via la transmission (Transmission). La puissance d'entraînement de ces deux sources peut être utilisée en fonction de conditions données (Figure 3-24). La batterie est chargée par la permutation du moteur électrique qui joue alors le rôle de génératrice. L'électricité de la batterie est utilisée pour entraîner les roues. Pendant le roulage, comme le moteur électrique joue à la fois le rôle de moteur et de génératrice, l'entraînement des roues ne peut pas se faire en même temps que le rechargement de la batterie.

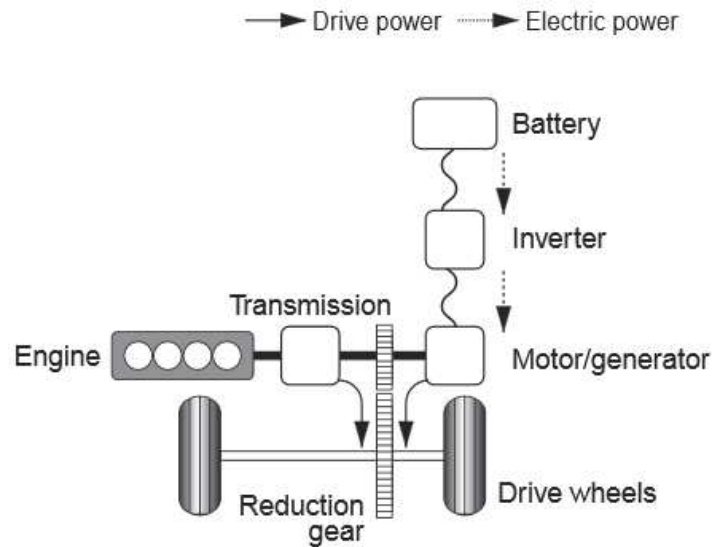


Figure 3-24 : Système hybride parallèle (Toyota Motor Corporation 2003)

4.1.3 Le système hybride mixte

Ce système combine le système hybride série avec le système hybride parallèle afin de maximiser les avantages des deux systèmes. En effet, cette combinaison permet grâce à un dispositif de répartition de la puissance (Power split device) de diviser la puissance du moteur thermique de sorte que le rapport de puissance s'applique directement sur les roues, le moteur électrique jouant également le rôle de générateur. Il y a deux moteurs et selon les conditions de route, le moteur thermique ou le moteur électrique peut entraîner les roues (Figure 3-25). De plus, lorsque cela est nécessaire, le moteur électrique peut entraîner les roues grâce à l'électricité fournie par la batterie, en même temps que celle-ci se recharge par la génératrice.

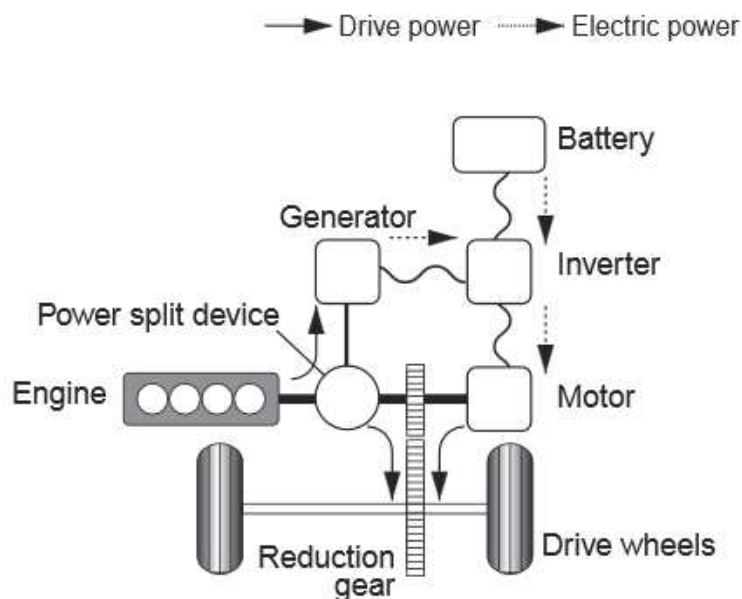


Figure 3-25 : Système hybride mixte (Toyota Motor Corporation 2003)

4.1 Mesure de la compacité réelle

Pour illustrer la mesure de compacité, nous disposons des trois architectures spatiales correspondant aux trois cas décrits précédemment.

Faute de données industrielles précises, nous avons supposé que dans les trois architectures, les dimensions et les formes simplifiées de la batterie (Battery), de l'onduleur (Inverter) et du moteur thermique (Engine) sont identiques pour chacune de ces architectures. Les autres composants (la génératrice (Generator), le moteur électrique (Motor), la transmission (Transmission) et le train épicycloïdal (Power split device)) sont utilisés et ont des dimensions et formes géométriques différentes selon l'architecture. Ces dimensions ont été estimées par un expert automobile.

La batterie ayant les mêmes dimensions quelle que soit l'architecture, et étant placée dans une autre zone du véhicule, elle ne sera plus considérée dans les architectures par la suite.

La Table 3-3 synthétise les informations géométriques des deux composants communs à toutes les architectures : moteur thermique et onduleur.

Composant	Forme	Dimensions (mm)
Onduleur (Inverter)	Parallélépipède	L = 400 l = 250 h = 100
Moteur thermique (Engine)	Parallélépipède	L = 500 l = 400 h = 680

Table 3-3: Dimensions des composants communes aux 3 architectures

4.1.1 Le système hybride série

L'architecture spatiale des composants géométriquement simplifiés (Table 3-4) du système hybride série est présentée sur la Figure 3-26.

Composant	Forme	Dimensions (mm)
Onduleur (Inverter)	Parallélépipède	L = 400 l = 250 h = 100
Moteur thermique (Engine)	Parallélépipède	L = 500 l = 400 h = 680
Génératrice (Generator)	Cylindre	r = 200 h = 200
Moteur électrique (Motor)	Cylindre	r = 200 h = 300

Table 3-4 : Dimensions et formes des composants du système hybride série

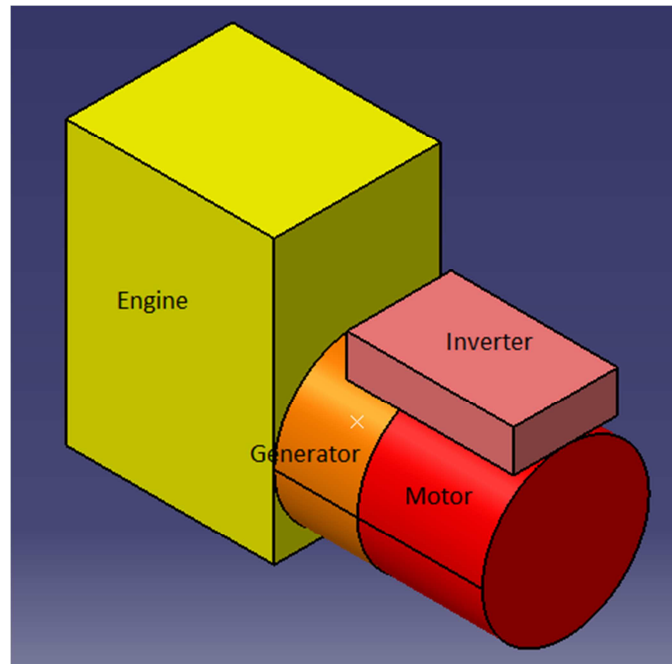


Figure 3-26 : Architecture d'un système hybride série

L'outil Catia permet d'extraire automatiquement dans un fichier texte les paramètres des composants.

Produit : Systeme_Hybride_Serie
Date : mardi, 13 mai 2014 16:26:15
Auteur : Aude

Component	Sub-Component	Aire[m2]	volume[m3]
Generator.1		0,503	0,025
Engine.1		1,624	0,136
Motor.1		0,628	0,038
Inverter.1		0,330	0,010
Systeme_Hybride_Serie		3,085	0,209

Figure 3-27 : Extraction des données de surface et de volume sous Catia pour le Système Hybride Série

Le volume réel du système est de 0,209m³.

Nous prenons arbitrairement comme hypothèse que le volume disponible est de 0,280m³.

Nous pouvons donc calculer l'indicateur de capacité réelle.

L'indicateur de capacité réelle est alors de

$$I_{CR} = \frac{\text{Volume réel de l'ensemble des composants}}{\text{Volume d'encombrement disponible}} = \frac{\sum V_i}{V_{ED}} = \frac{0,209}{0,280} = 0,746$$

L'assemblage occupe 74,6% du volume disponible.

4.1.2 Le système hybride parallèle

L'architecture spatiale du système hybride parallèle est donnée sur la Figure 3-28 et les dimensions et formes des composants sont données dans la Table 3-5.

Composant	Forme	Dimensions (mm)
Onduleur (Inverter)	Parallélépipède	L = 400 l = 250

		h = 100
Moteur thermique (Engine)	Parallélépipède	L = 500 l = 400 h = 680
Transmission (Transmission)	Cylindre	r = 200 h = 250
Moteur électrique et Génératrice (Motor/Generator)	Cylindre	r = 200 h = 350

Table 3-5 : Dimensions et formes des composants du système hybride parallèle

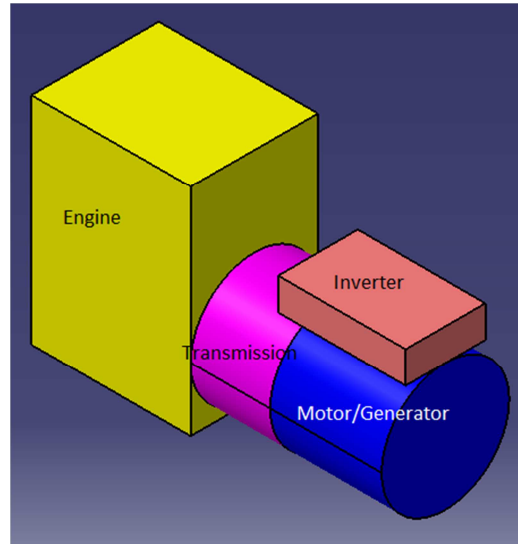


Figure 3-28 : Architecture d'un système hybride parallèle

Comme précédemment, nous pouvons extraire du logiciel Catia le volume des composants et de l'assemblage :

Produit : Systeme_Hybride_Parallele			
Date : mercredi, 14 mai 2014 10:43:25			
Auteur : Aude			
Composant	Sub-Component	Aire[m2]	volume[m3]
Transmission.1		0,565	0,031
Engine.1		1,624	0,136
Motor/Generator.1		0,691	0,044
Inverter.1		0,330	0,010
systeme_Hybride_Parallele		3,211	0,221

Figure 3-29 : Extraction des données de surface et de volume pour le système hybride parallèle

Le volume total réel occupé par les composants de ce système hybride parallèle est de 0,221m³. Gardons l'hypothèse que le volume disponible est de 0,280m³, dans ce cas, l'indicateur de compacité réelle est de :

$$I_{CR} = \frac{\text{Volume réel de l'ensemble des composants}}{\text{Volume d'encombrement disponible}} = \frac{\sum V_i}{V_{ED}} = \frac{0,221}{0,280} = 0,789$$

L'assemblage occupe donc 78,9% du volume disponible.

4.1.3 Le système hybride mixte

L'architecture spatiale du dernier système : hybride mixte, est détaillée sur la Figure 3-30 et les caractéristiques physiques des composants sont données dans la Table 3-6.

Composant	Forme	Dimensions (mm)
Onduleur (Inverter)	Parallélépipède	L = 400 l = 250 h = 100
Moteur thermique (Engine)	Parallélépipède	L = 500 l = 400 h = 680
Génératrice (Generator)	Cylindre	r = 200 h = 200
Moteur électrique (Motor)	Cylindre	r = 200 h = 300
Répartiteur de puissance (Split)	Cylindre	r = 35 h = 120

Table 3-6 : Dimensions et formes des composants du système hybride mixte

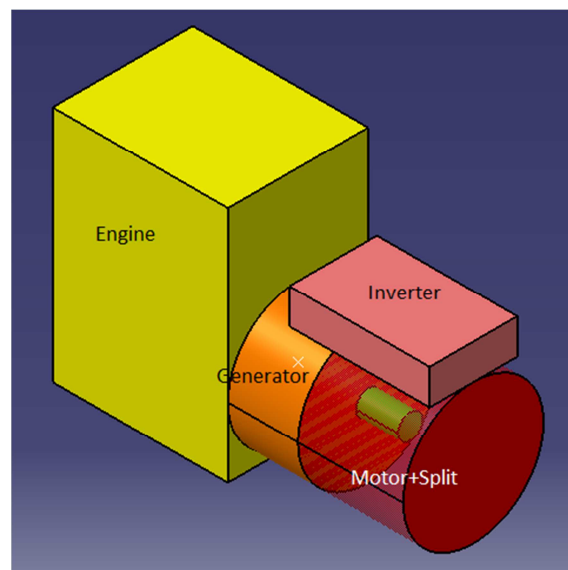


Figure 3-30 : Architecture d'un système hybride mixte

Voici les volumes de chaque composant :

Produit : Systeme_Hybride_mixte Date : mercredi, 14 mai 2014 10:27:38 Auteur : Aude			
Component	Sub-Component	Aire[m2]	volume[m3]
Generator.1		0,503	0,025
Engine.1		1,624	0,136
Split.1		0,034	0,000461814
Motor.1		0,628	0,038
Inverter.1		0,330	0,010
Systeme_Hybride_mixte		3,119	0,209

Figure 3-31 : Extraction des données de surface et de volume du système hybride mixte

Le volume total réel du système est alors de 0,209 m³. Le volume disponible étant de 0,280m³, nous pouvons calculer l'indicateur de compacité réelle :

$$I_{CR} = \frac{\text{Volume réel}}{\text{Volume d'encombrement disponible}} = \frac{\sum V_i}{V_{ED}} = \frac{0,209}{0,280} = 0,746$$

L'assemblage occupe 74,6% du volume disponible.

4.1.4 Synthèse

Le tableau (Table 3-7) ci-dessous résume les volumes réels pour chaque architecture ainsi que le taux d'occupation réelle correspondant.

Architecture du système hybride	Nombre de composants	Volume réel	Volume disponible	Taux d'occupation réelle
Série	4	0,209	0,280	0,746
Parallèle	4	0,221	0,280	0,789
Mixte	5	0,209	0,280	0,746

Table 3-7 : Synthèse des résultats de calculs de l'indicateur de compacité réelle

Nous pouvons constater qu'entre la structure série et mixte l'indicateur de compacité réelle est le même. Il y a un composant de plus (répartiteur de puissance) dans l'architecture mixte mais son volume est négligeable par rapport aux autres composants ce qui explique que ces deux architectures aient la même valeur d'indicateur de compacité réelle. En effet, l'indicateur de compacité réelle ne prend pas en compte le nombre de composants. Pour cela, il sera judicieux de coupler l'indicateur d'intégration multi-niveau (détaillée au Chapitre 2 §3.1.2) avec l'indicateur de compacité, dans les perspectives de ces travaux.

De plus, l'architecture parallèle a une valeur d'indicateur de compacité réelle plus élevée que les deux autres. A volume disponible égal, il y a donc moins de place inoccupée et à la lecture du seul indicateur de compacité réelle, il ressort qu'il y a moins d'espace perdu ou moins de place pour rajouter des composants, suivant l'objectif du concepteur.

4.2 Indicateur de forme

Prenons l'hypothèse que le volume disponible soit un parallélépipède rectangle de dimension 1000 x 400 x 700. Nous venons de voir, grâce à l'indicateur de compacité réelle que les trois assemblages étaient « idéalement intégrables » dans le volume disponible. Maintenant pour savoir si ces assemblages sont contenus ou non dans la forme du volume d'encombrement disponible, nous devons, d'après l'équation (3.7) ou (3.9), calculer la demi-distance d'Hausdorff de la forme du volume disponible à l'assemblage.

Pour l'architecture spatiale du système hybride série et celle de l'architecture hybride mixte, la valeur de l'indicateur de forme est « vrai ». En effet, la forme de l'assemblage de ces deux architectures est plus petite que la forme du volume disponible.

En revanche, la demi-distance entre la forme du volume disponible et la forme de l'assemblage du système hybride parallèle, calculée manuellement, vaut 100 mm. Le résultat de l'indicateur de forme est donc « faux ». Cela signifie donc que les composants du système hybride parallèle ne pourront pas être assemblés selon l'architecture spatiale présentée.

Néanmoins gardons à l'esprit que dans la mesure où nous sommes en phase de préconception avec des volumes simplifiés et des données géométriques simplifiées, il faudrait donc, en toute rigueur, évaluer cette distance en prenant en compte l'incertitude liée à l'approximation de simplification géométrique, pour pouvoir en évaluer sa pertinence. La prise en compte des incertitudes des données sera une des pistes de poursuite de ces travaux.

L'objectif à termes est d'implémenter dans un outil CAO 3D (pour connaître le placement des composants), l'équation de cette distance afin d'effectuer le calcul automatiquement. Cette partie sera évoquée au §5.4.

4.3 Indicateur de compacité convexe

L'indicateur de compacité convexe (3.11) est fonction du volume réel des composants et du volume convexe de l'assemblage. Pour calculer cet indicateur, il faut au préalable pouvoir construire et calculer le volume de l'enveloppe convexe de l'assemblage des composants.

La définition de l'enveloppe convexe est donnée par :

Soit A une partie de E, l'enveloppe convexe de A est l'intersection de toutes les parties convexes (§3.1.4.1) de E qui contiennent A.

L'enveloppe convexe d'un assemblage est parfois difficile à obtenir pour certaines formes complexes, surtout en 3D.

Une première piste de développement consiste à rechercher le meilleur algorithme et l'implémenter dans le modèleur Catia en programmation VB (Visual Basic).

Après avoir cherché dans la littérature des algorithmes de calcul d'enveloppe convexe (Preparata and Shamos 1985; Clarkson and Shor 1988; Clarkson and Shor 1989; Berg et al. 2000) nous avons retenu une méthode consistant à rechercher de manière incrémentale les faces triangulaires qui constituent l'enveloppe. La méthode⁴⁶ est détaillée en Annexe 6.

Un exemple d'une autre méthode permettant de construire l'enveloppe convexe en langage Java est également décrite en Annexe 7.

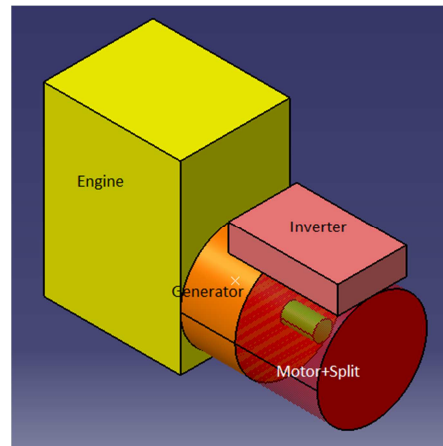
Pour construire l'enveloppe convexe, les composants doivent être au préalable discrétisés. Cette étape est réalisable avec l'application VB du logiciel Catia.

Cependant, faute de temps, les précédentes méthodes et les algorithmes identifiés n'ont encore pas pu être implémentés. Ce travail fera donc partie des perspectives...

4.4 Indicateur d'accessibilité

Dans notre exemple, une seule architecture fait apparaître un composant à l'intérieur d'un autre : l'architecture hybride mixte. Ci-dessous, un rappel de la Figure 3-30 :

⁴⁶<http://www.enseignement.polytechnique.fr/informatique/profs/Georges.Gonthier/pi98/env3D.html#Preparata-Shamos>



Le moteur électrique (motor) devient le composant enveloppe, cependant il n'y a pas d'accès au répartiteur de puissance (split) puisque ce composant se trouve entre l'onduleur et le moteur électrique sans passage d'accès. L'indicateur d'accessibilité pour le répartiteur de puissance est alors égal à :

$$I_a = \left(1 - \frac{r_{objet}}{r_i \text{ de tous les passages}}\right) \times \frac{0}{0} = 0$$

Cette information est capitale pour le concepteur, qui peut ainsi attirer l'attention des équipes du bureau d'études, sur les aspects d'assemblage, de maintenabilité, de connectiques...

5 Implémentation des données géométriques en SysML

Pour répondre au problème d'intégration physique des systèmes mécatroniques, une métrique d'intégration physique est nécessaire afin d'évaluer les différentes architectures physiques candidates (suivant leur compacité, leur volume restant disponible et leur accessibilité).

Dès lors, il est nécessaire d'avoir accès en amont aux données de volumes, de surfaces, de distances, de formes des composants dans le système. Ainsi, l'introduction du positionnement géométrique dans une approche MBSE, bien avant la phase de conception détaillée (où il est habituellement réalisé avec des outils de CAO), permettra :

- A l'architecte système d'extraire les bonnes spécifications géométriques dès le début et donc de réduire le temps passé sur la conception en limitant le nombre d'itérations,
- de fournir un moyen graphique de partager de façon concise et intelligible des informations géométriques entre les équipes multidisciplinaires de l'ingénierie mécatronique,
- d'assurer une traçabilité et la cohérence des données géométriques peu coûteuses entre les premières phases (les exigences géométriques) et les dernières phases (géométrie détaillée 3D) de la conception.

Ainsi, au travers de l'implémentation de ces métriques au niveau des architectures du modèle « Système », l'objectif est de prendre en compte dès que possible la géométrie des composants dans le cycle de conception.

Nous avons choisi le langage SysML (System Modeling Language)⁴⁷ pour implémenter et expérimenter cette approche. Pour cela, nous avons ainsi défini un profil géométrique spécifique permettant d'enrichir, en phase amont, le modèle « Système », avec la géométrie (forme et dimensions) de chaque composant.

5.1 SysML et les profils

Bien que le langage SysML ^{Erreur ! Signet non défini.} décrit au Chapitre 1 §1.2.2.2 soit de plus en plus un sujet d'étude et de développement pour l'ingénierie des systèmes quel que soit le domaine (Holt and Perry 2008), il n'a pas encore été mis en œuvre pour la modélisation géométrique des premiers stades de la conception. Pourtant, pour les systèmes mécatroniques, les contraintes de positionnement des composants sont primordiales (Chen et al. 2009; Moullec et al. 2013) pour tenir compte de la compacité (Reynaerts et al. 1998; Grossard et al. 2013).

Par définition, *un profil contient des mécanismes permettant d'étendre les métaclasse d'un métamodèle. Il contient un ensemble de stéréotypes, de contraintes et d'extensions de diagrammes, qui sont utilisés pour adapter le langage UML à une application ou à un domaine particulier* (Turki 2008b). Ainsi, SysML peut être considéré comme étant un profil UML adapté au domaine de l'ingénierie système (Holt and Perry 2008). Un profil d'un profil existant est appelé une extension. Aussi, nous proposons d'étendre le profil SysML à la géométrie.

La valeur ajoutée de cette extension géométrique a plusieurs objectifs :

- Fournir un moyen à l'architecte système de spécifier les exigences géométriques pour enrichir les architectures physiques ;
- Prendre en compte les contraintes géométriques (positionnement des composants), pour faciliter le travail des équipes de conception préliminaire, en pré-positionnant les composants, avant d'évaluer leurs interactions physiques correspondantes (travaux de Romain BARBEDIENNE (Barbedienne et al. 2014)) ;
- Implémenter des métriques géométriques pour évaluer l'intégration physique (compacité, volume disponible, distances d'interaction physique (Barbedienne et al. 2014), ...), lors de la conception des systèmes mécatroniques.

5.2 Définition de l'extension SysML pour la géométrie

L'architecture physique décrit les composants constituant le système. Cette architecture est généralement représentée, en SysML, dans un diagramme de définition de bloc (BDD). Les composants physiques sont alors définis, et par conséquent les caractéristiques techniques liées aux composants choisis sur étagère ou spécifiés sont connues. Pour calculer des métriques géométriques à partir de ces architectures, il faut au préalable qu'un certain

⁴⁷<http://www.omg.sysml.org>

nombre de données géométriques (forme, dimensions et positions des composants) soient ajoutées aux entités de ces modèles, au travers de la nouvelle extension SysML développée.

Cette extension a été définie en utilisant Artisan Studio (v7.4) d'Atego⁴⁸. Elle permet la modélisation des systèmes mécatroniques hautement intégrés, en prenant en compte les contraintes géométriques associés.

La Figure 3-32 présente une architecture physique dont les caractéristiques géométriques et physiques ont été ajoutées comme attributs aux blocs « component » relatifs aux composants de l'architecture.

Le bloc « System » récupère la valeur de la métrique correspondant à cette architecture, définie à l'aide d'un diagramme paramétrique (Figure 3-40).

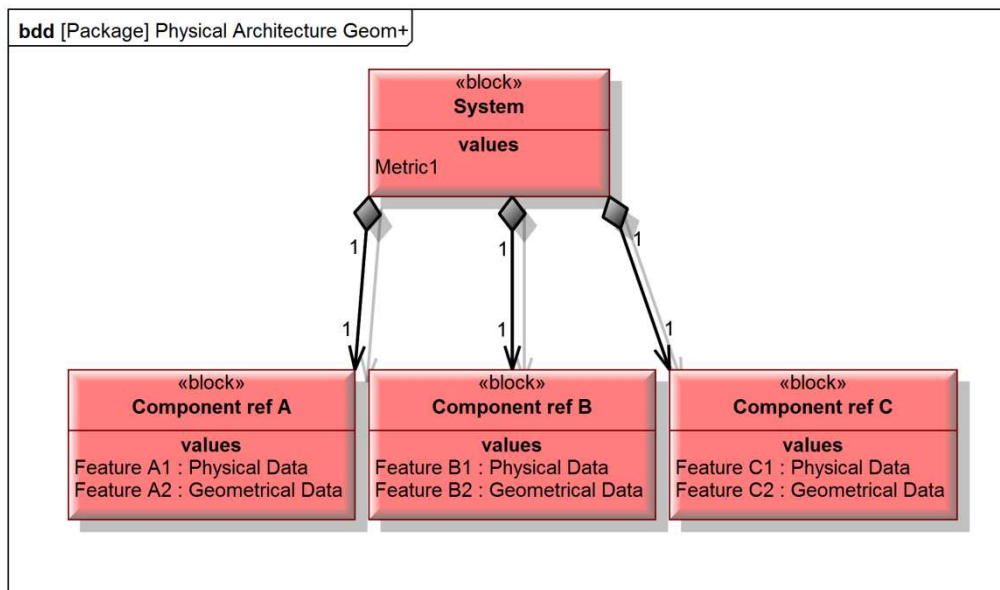


Figure 3-32 : Composants enrichis de paramètres géométriques

Par ailleurs, cette extension définit des blocs stéréotypés « Geometry » pour chaque géométrie simplifiée, ainsi que les paramètres de définition de géométrie associés. Une librairie de volumes simplifiés a été développée, avec une interface graphique utilisateur pour faciliter sa prise en main et sa mise en œuvre (Figure 3-33). Cette librairie comprend la sphère, le cylindre, le cylindre creux, les parallélépipèdes (régulier, creux, quelconque), le cône, le prismatique, le prismatique creux, le tore.

⁴⁸ <http://www.atego.com/products/atego-modeler/>

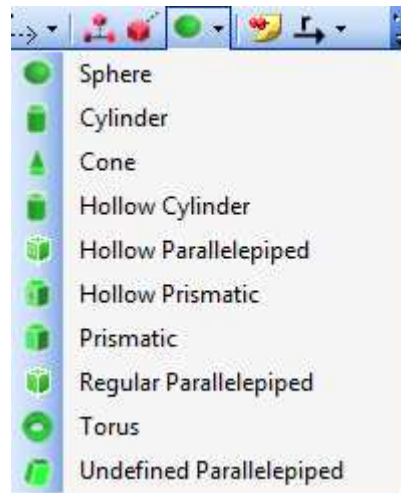


Figure 3-33 : Librairie des composants géométriques

Cette librairie contient tous les modèles et éléments spécifiques de géométrie associés à chaque forme : les dimensions maximales désirées ou connues, et le calcul et / ou spécification de la surface et du volume. La Figure 3-34 présente un exemple de ces composants.

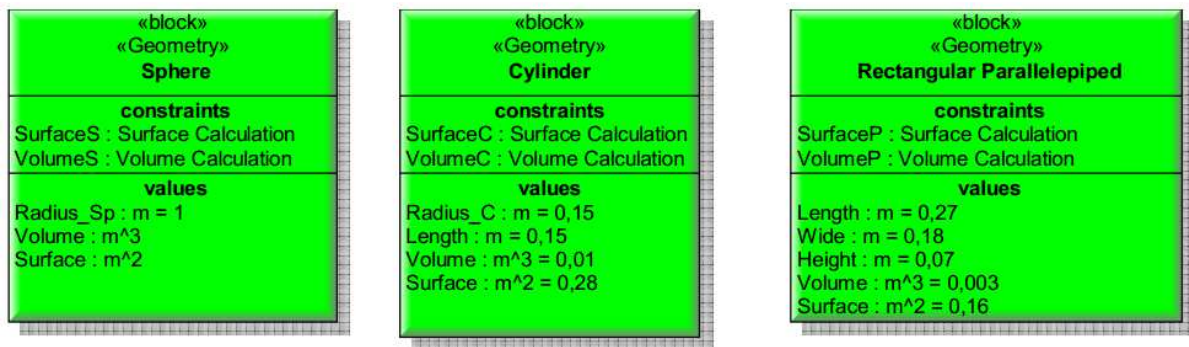


Figure 3-34 : Exemple de composants géométriques avec leurs paramètres

Enrichissement de l'architecture physique

Pour chaque bloc d'une architecture logique ou physique, défini comme un composant, il est alors très simple d'appliquer un stéréotype « géométrique » (clic droit sur « Applied stereotypes » \ Geometry) pour pouvoir utiliser les éléments implémentés dans l'extension géométrique.

Deux scénarios sont possibles pour spécifier la géométrie des composants.

Scénario N°1 : le concepteur crée son architecture de composants à partir d'éléments géométriques. Par exemple pour une architecture logique, le concepteur a identifié les organes/technologies sans connaître les composants détaillés. Il peut définir des volumes enveloppes d'encombrement par fonction ainsi que sa géométrie simplifiée envisagée comme une sphère, un cylindre, etc. Pour cela, le concepteur peut créer un bloc géométrique et lui associer une forme géométrique comme la sphère, le cylindre à l'aide du menu contextuel (Figure 3-35), ou il peut prendre directement un composant géométrique de la librairie dans la liste déroulante (Figure 3-33).

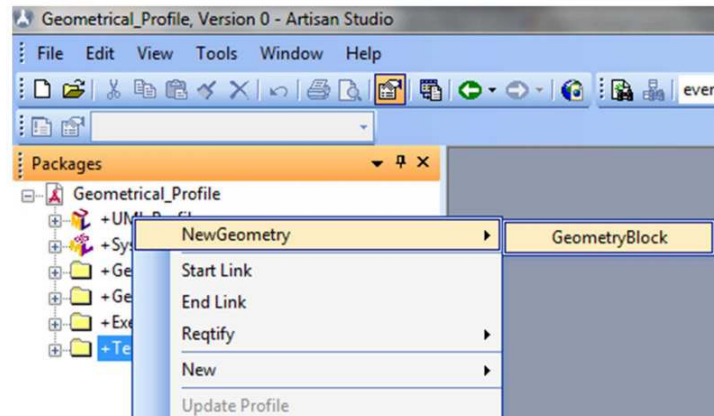


Figure 3-35 : Création d'un bloc géométrique

Scénario N°2 : Le concepteur a déjà une architecture physique existante avec des composants réels. Il souhaite l'enrichir avec une dimension géométrique en appliquant le stéréotype géométrique sur ces composants réels. Pour cela, à l'aide d'un clic droit sur le composant, il lui associe simplement la géométrie désirée (Figure 3-36), faisant apparaître dans le composant tous les attributs correspondants à cette forme (Figure 3-34).

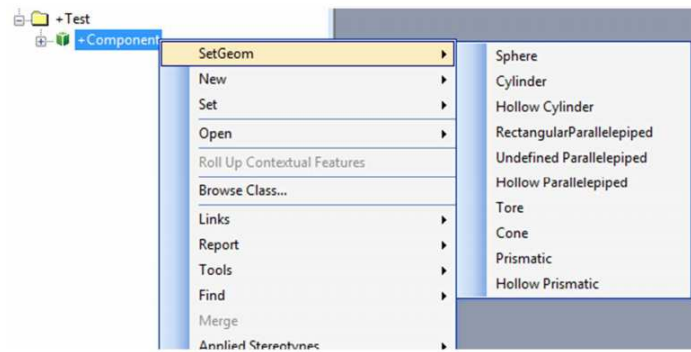


Figure 3-36 : Choix d'une forme spécifique sur un composant existant stéréotypé « geometry »

Enfin, la modélisation simplifiée de la géométrie, implémentée dans SysML avec cette extension géométrique va permettre au concepteur de construire et de calculer des métriques pour faciliter le choix entre plusieurs architectures candidates, conformément à un objectif d'intégration physique.

Pour cela, il est nécessaire de modéliser la métrique précédemment décrite en s'appuyant sur les paramètres géométriques des composants définis dans le BDD, à l'aide d'un diagramme paramétrique. Celui-ci permet d'intégrer l'analyse du système via les métriques, en les formalisant par une expression mathématique liée aux paramètres géométriques du système : comme le volume, les dimensions du composant, etc (Figure 3-34).

5.3 Applications aux architectures spatiales de la motorisation hybride

Les architectures des trois architectures de motorisation hybrides sont présentées sur les figures suivantes :

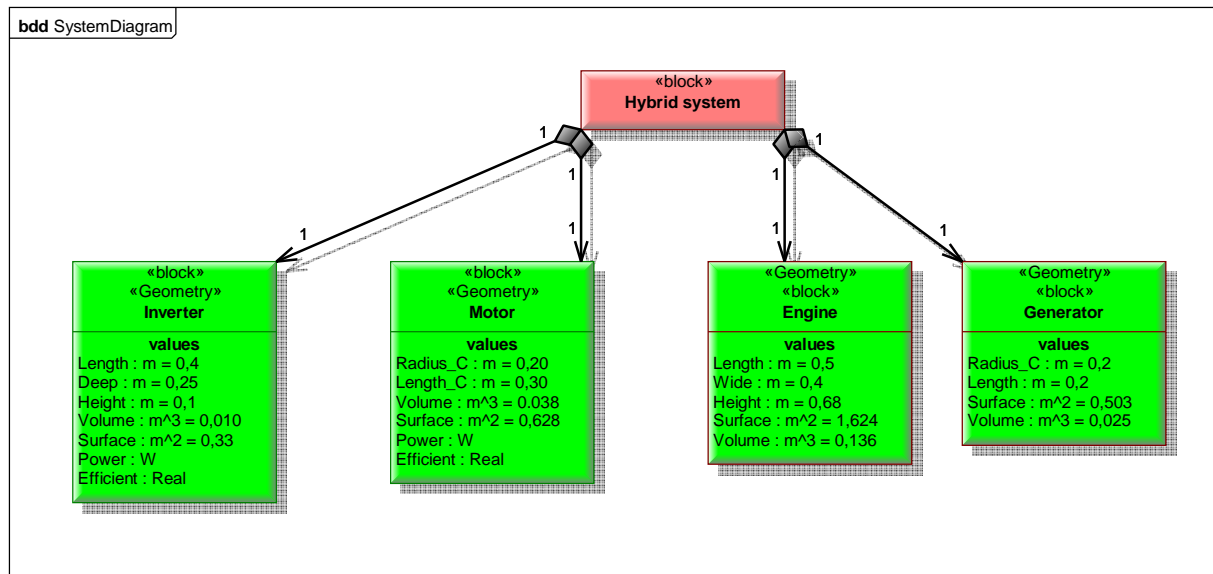


Figure 3-37 : Architecture du système hybride série

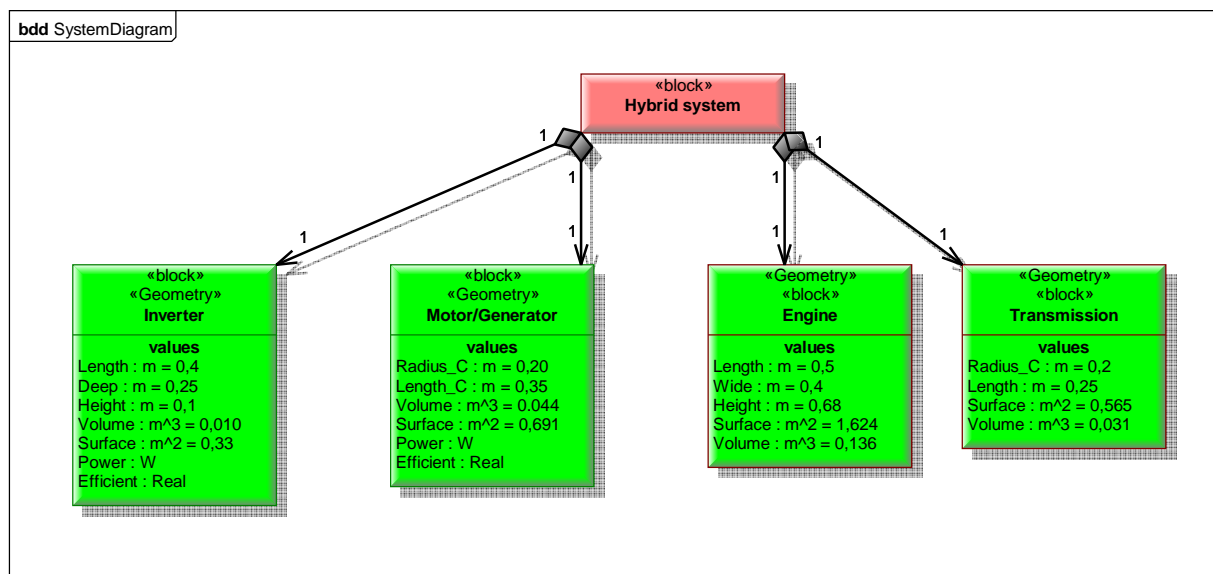


Figure 3-38 : Architecture du système hybride parallèle

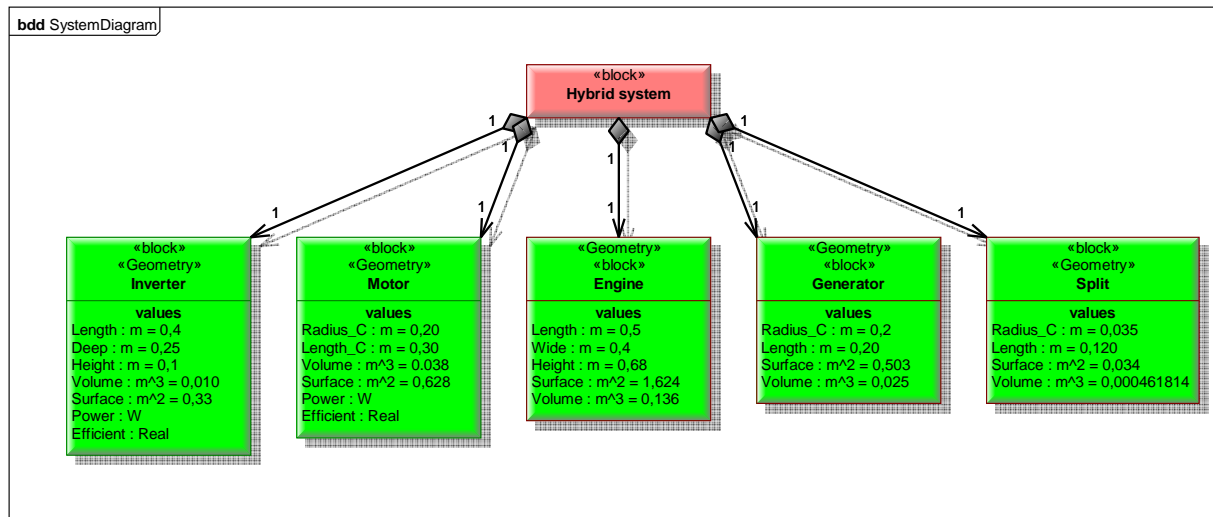


Figure 3-39 : Architecture du système hybride mixte

Le volume et la surface de chaque composant est calculée par un diagramme paramétrique couplé au solveur parasolver⁴⁹.

Le diagramme paramétrique de l'indicateur de compacité réelle pour l'architecture série est donné à titre d'illustration (Figure 3-40).

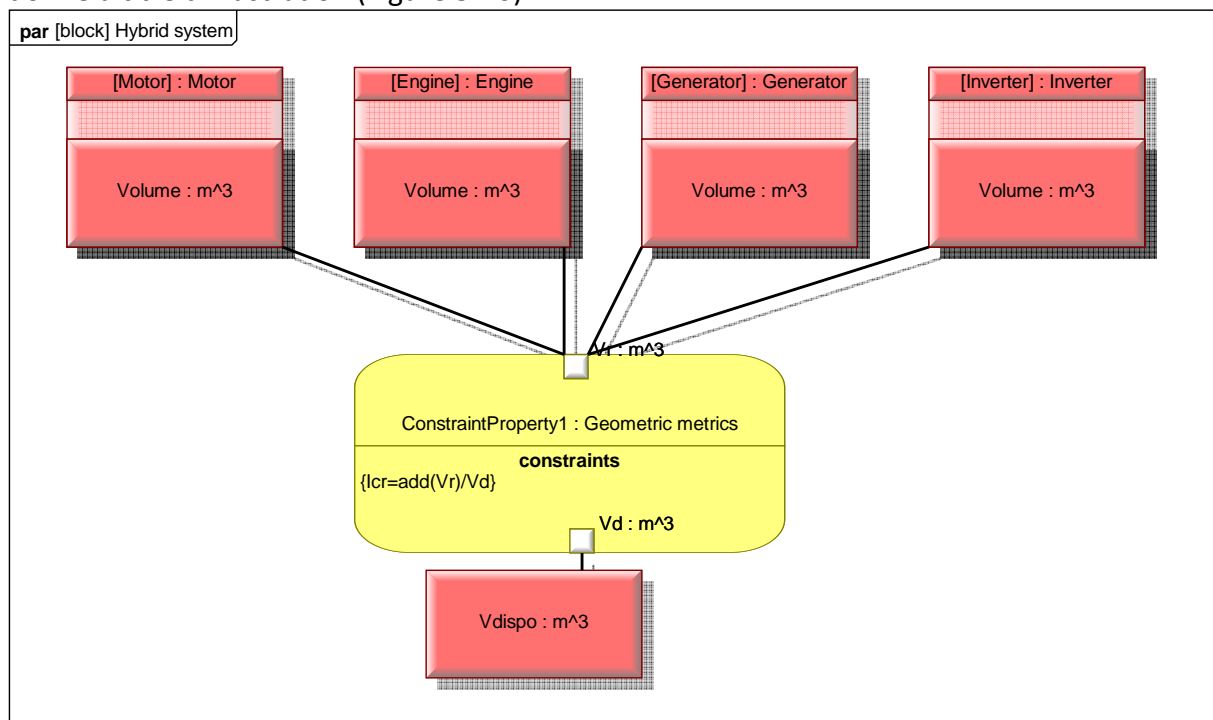


Figure 3-40 : Diagramme paramétrique permettant de calculer la métrique lcr

5.4 Placement des composants

Pour calculer l'indicateur de forme et l'indicateur de compacité convexe qui requièrent la position des composants, nous avons besoin de rajouter celle-ci à l'extension SysML que nous avons développée. Pour cela, nous avons envisagé d'intégrer la méthode des SATT décrite dans l'état de l'art au §2.2. Ces travaux seront réalisés par la suite dans le cadre d'une collaboration avec les travaux de thèse de Romain Barbedienne.

⁴⁹ <http://www.atago.com/campaigns/artisan-studio-parasolver-2/>

6 Discussion/perspectives

Dans l'exemple mécatronique choisi pour l'expérimentation, les valeurs sont très proches et ne permettent pas de faire une comparaison des architectures avec des valeurs significatives sur l'indicateur de compacité réelle. En effet, sur les trois architectures, seule une dispose d'un composant supplémentaire, mais de volume trop négligeable par rapport aux autres composants.

Le développement d'une interface avec un outil CAO devrait permettre de calculer les autres indicateurs prenant en compte le positionnement des composants, et ainsi de faire apparaître un écart sur la valeur de la métrique globale d'intégration physique plus significatif.

Il faudra également naturellement la vérifier sur des exemples industriels plus complexes.

Nous avons vu que les métriques d'intégration physique faisaient intervenir le positionnement des composants. Il semble donc intéressant, même en préconception, de pouvoir disposer d'un outil CAO simplifié (volumes simples, contraintes de position basiques) pour permettre à l'architecte système ou à l'architecte modèle de positionner rapidement les différents composants, pour pouvoir évaluer les différentes architectures physiques spatiales possibles, à l'aide de métriques géométriques. Ainsi l'intégration physique des systèmes mécatroniques pourra être évaluée, en prenant en compte l'accessibilité, les couplages multi-physiques, ... Ces travaux sont en cours dans le cadre de la thèse de Romain Barbedienne, ils seront naturellement couplés à de nouvelles métriques à développer.

Enfin, concernant la distance d'Hausdorff qui permet de mesurer la distance entre deux sous-ensembles, il nous apparaît intéressant d'étendre l'utilisation de cette distance à la mesure de couplages multi-physiques. En effet, nous pouvons imaginer de mesurer la distance d'interaction physique entre deux composants au-delà de laquelle les couplages multi-physiques seraient négligeables. Cette idée devrait nous permettre de développer des métriques de couplages multi-physiques.

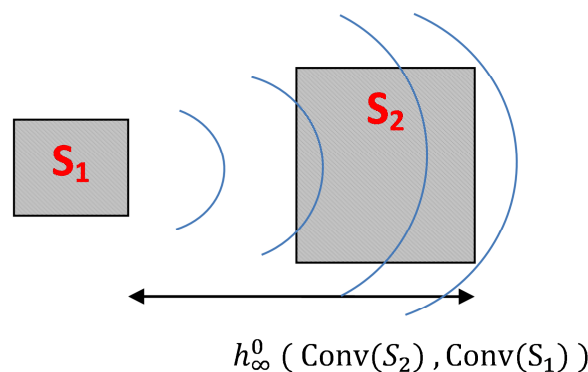


Figure 3-41 : Utilisation de la distance d'Hausdorff pour le placement des composants sous l'influence physique d'un autre composant

7 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'intégration physique dans la conception des systèmes mécatroniques, et plus particulièrement à la métrique permettant à l'architecte système de faire ses choix d'architectures en fonction de cet objectif.

L'architecte système pourra alors prendre en compte les contraintes géométriques et orienter ses choix d'architectures sans attendre la phase de conception détaillée.

Pour cela, nous avons proposé quatre indicateurs pour construire la métrique d'intégration physique (MIP) : deux de compacité, un de forme et un d'accessibilité. Après en avoir donné leur définition, nous avons présenté les différents indicateurs et leur mode de calcul, avant d'expérimenter l'indicateur de compacité réelle sur l'exemple d'un système de motorisation hybride, grâce à l'automatisation des calculs de volume dans le logiciel Catia V5.

Les travaux de cette thèse portent sur la phase de préconception, où les données nécessaires pour calculer ces indicateurs ne sont pas toujours disponibles pour l'architecte système. Ainsi, selon les données (volume, forme de l'encombrement et de l'assemblage) dont il dispose à un instant donné et ses objectifs d'intégration, l'architecte utilisera plutôt tel ou tel indicateur, pour faire son choix.

Pour finir, le calcul de ces métriques en phase de préconception suppose que l'architecte système dispose d'informations géométriques dans le modèle système. Pour cela, nous avons développé un profil géométrique grâce à une extension du langage SysML permettant d'enrichir les modèles avec les données géométriques connues. Cette extension permet ainsi de calculer les indicateurs pour chaque architecture modélisée, et ainsi de justifier et tracer les choix de conception.

Les principales perspectives de ces métriques concernent l'implémentation du calcul automatique des indicateurs nécessitant la prise en compte de la position des composants, et donc un interfaçage avec un outil 3D.

Chapitre 4

Processus de création de métriques personnalisées

1	Introduction.....	165
1.1	Contexte industriel.....	165
1.2	Le concept	166
1.3	Règles de formalisation d'une métrique.....	168
2	Description du processus	168
2.1	L'analyse des besoins	168
2.2	Les objectifs à évaluer	169
2.3	Identification des paramètres influents.....	171
2.4	Construction de la métrique	172
2.5	Validation de la métrique.....	173
2.6	Analyse de la sensibilité	173
3	Application de la méthode	175
3.1	L'analyse du besoin	175
3.2	Définition de l'objectif.....	175
3.3	Identification des paramètres influents.....	176
3.4	Construction de la métrique	176
3.5	Validation de la métrique.....	179
4	Discussion et perspectives	179
5	Conclusion	181

Introduction

Après avoir proposé des métriques pour évaluer l'intégration mécatronique et échangé avec des industriels de systèmes mécatroniques, nous nous sommes rendus compte que finalement, et selon la culture de l'entreprise, son organisation, sa stratégie ou l'expérience des concepteurs, les métriques utilisées pour évaluer leurs produits étaient fort différentes. De ce fait, il était donc difficile de fournir une métrique universelle, même pour un objectif donné, dans un même secteur industriel donné, qui puisse convenir à tous les concepteurs de toutes les entreprises.

Ce chapitre présente une démarche afin de générer des métriques personnalisées, considérant les besoins de l'entreprise tout en respectant les règles de formalisation des métriques définies au chapitre § 3.2.1.

Pour conclure ce chapitre et après avoir évoqué l'utilisation actuelle des métriques par les industriels, nous présentons en détail chaque étape du processus développé, avec un bref rappel des règles requises de formalisation des métriques. Pour terminer, le processus est appliqué à l'objectif de modularité.

1.1 Contexte industriel

Bien que des métriques soient déjà largement développées et utilisées industriellement, en particulier pour le domaine logiciel, et ce depuis des décennies (Côté et al. 1988), les métriques existantes détaillées au Chapitre 1 §4 sont généralement spécifiques à un seul domaine.

Etant donné que les systèmes mécatroniques sont des systèmes intégrés, multifonctions, multi-domaines, multi-physiques, ils présentent tout au long du cycle de vie, une multi-vue complexe qu'il est difficile à analyser.

En effet, la conception de ces systèmes doit être commune à toutes les équipes techniques quel que soit leur domaine d'application (Electronique, Automatique, Mécanique, Informatique...), pour permettre une conception intégrée optimale. Ainsi, le besoin pour les concepteurs de ces systèmes d'acquies un moyen d'évaluation qui leur soit spécifique devient donc crucial.

Rappelons que, le quotidien des équipes de conception consiste à proposer des solutions répondant à leur cahier des charges, et à faire par la suite des compromis avec l'ensemble des équipes techniques, ayant la plupart du temps des objectifs contradictoires. Dès lors, il est souhaitable d'identifier un moyen pour faciliter ces compromis pour les différents choix de conception.

Les métriques sont un support d'aide à la décision intéressant pour aider à formaliser et tracer les critères sur lesquels les décisions finales seront prises. Elles permettent notamment de justifier les solutions retenues et de tracer les solutions écartées tout en facilitant la compréhension des décisions prises. La difficulté principale reste donc à trouver la « bonne » métrique qui répondra à un objectif de conception fixé et sera représentative du « vécu » de l'entreprise.

Actuellement, suivant que l'entreprise de produits mécatroniques est organisée par services de compétences (formation mécanique, électronique, informatique...) ou par technologie de produits (actionneurs, capteurs, câbles...), voire aussi par niveau de conception (exigences, analyse fonctionnelle, conception préliminaire, détaillée...), les métriques utilisées pour valider la faisabilité d'un concept n'adresseront pas les mêmes prises de décision.

A titre d'exemple, considérons la métrique d'intégration multi-domaine développée au Chapitre 2 §3.2. Si l'entreprise est organisée par service de compétences, cette métrique permettra d'estimer les contraintes de planning pour les jalons de choix de conception entre les équipes, mais aussi la complexité (donc les ressources nécessaires correspondantes) liée à l'interaction nécessaire entre des équipes aux méthodes, outils, contraintes et objectifs différents. Il s'agit notamment du besoin en formation des équipes pour qu'elles puissent échanger avec un langage commun, des connaissances de base dans les autres domaines impliqués pour en connaître les enjeux, et de l'accompagnement de proximité pour le chef de projet en charge de la cohérence de développement...

Si la structure a une organisation par technologie de composants (tous domaines confondus), cette métrique permettra d'identifier au sein du même composant (même service) les risques de couplages multi-physiques liés à la proximité de sous-composants multi-domaines. Une structure par niveau de conception, pourra utiliser cette métrique, pour évaluer le besoin collaboratif d'échanges de données de différents domaines dans chacune des phases de conception, et ainsi décider de la méthode/cycle de conception le plus approprié.

De manière similaire, si la stratégie de l'entreprise est de sous-traiter une partie de ces composants, il peut être intéressant de connaître le niveau d'intégration multi-domaine du système à sous-traiter pour cibler les compétences, la taille et le type de fournisseurs.

Ces considérations nous ont naturellement amenés à réfléchir à un processus de construction de métriques personnalisées, qui seront adaptées à la structure, aux produits et aux processus des industriels, et notamment du secteur mécatronique, ce dernier devant faire face à une complexité des produits particulièrement élevée.

1.2 Le concept

L'inconvénient des métriques existantes proposées par d'autres auteurs (Chapitre 1 §4) est que généralement elles « ne mesurent pas les bonnes choses de la bonne manière afin de prendre les bonnes décisions au bon moment » (Smith 2013). Par conséquent, l'idée est de proposer aux industriels de systèmes mécatroniques, un processus leur permettant de construire leurs propres métriques pour faciliter la prise de décisions lors de la conception de leurs systèmes.

La démarche représentée sur la Figure 4-1 est une première approche permettant de construire des métriques personnalisées, afin d'évaluer un processus, un projet, une conception du système et une qualité du produit, adressant des objectifs de conception fixés.

Après l'analyse des besoins des parties prenantes, un ou plusieurs « objectifs de conception » à évaluer sont définis. A ce propos, J.R. Smith (Smith 2013) explique que pour l'industrie des technologies d'informations (IT), les objectifs stratégiques sont

habituellement une reconnaissance mondiale, le leadership technologique, l'innovation et la satisfaction de la communauté.

S'agissant de la mécatronique, les objectifs seront plutôt l'intégration, la robustesse, la maintenabilité et la fiabilité. Une *librairie d'objectifs* peut également proposer des objectifs mécatroniques courants ou extraits d'une base de données interne réalisée à partir de l'expérience de l'entreprise.

Aussi, l'objectif de la métrique ayant été défini, le concepteur de métriques peut alors en identifier les paramètres influents grâce à son savoir-faire et son expérience, ou à partir d'une *librairie de paramètres* et/ou d'une analyse de la sensibilité. La métrique désirée peut alors être construite à partir des paramètres influents identifiés précédemment. La métrique conçue est une formulation mathématique. Elle peut être construite à partir de fonctions mathématiques simples comme les opérateurs élémentaires ou des équations plus complexes. Elle doit refléter le ressenti de l'entreprise, pour cela le concepteur s'appuiera sur des scénarios de cas réels pour lesquels il connaît la valeur attendue.

Il peut également s'appuyer sur une *librairie de métriques* et sur une *librairie de fonctions mathématiques prédéfinies*.

Si la formulation mathématique de l'étape précédente est trop difficile à établir ou si la validation de la métrique par un test sur un autre scénario industriel n'est pas avérée, une analyse de la sensibilité est recommandée. Cette analyse permet alors de corriger à la fois les paramètres influents et la formulation mathématique de la métrique afin de l'améliorer jusqu'à l'obtention d'une métrique adéquate.

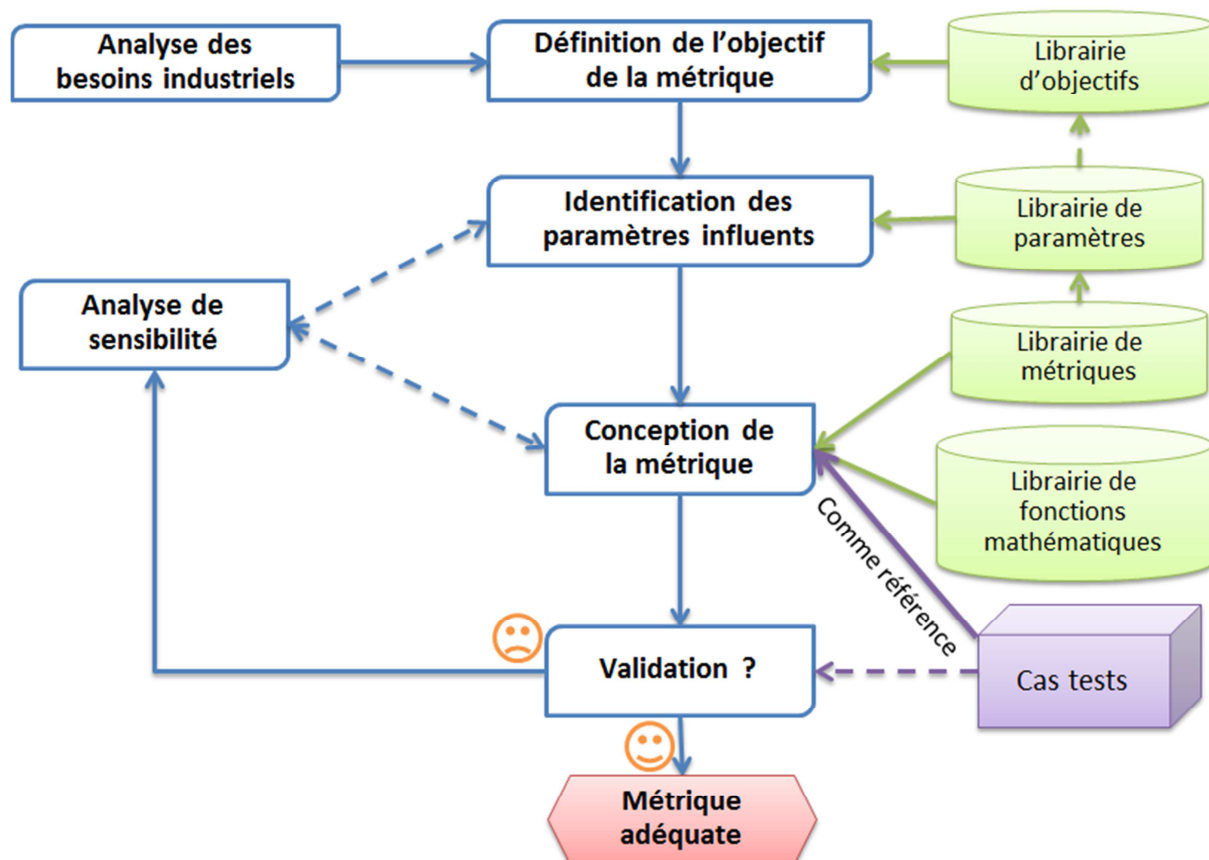


Figure 4-1: Processus de "l'usine à métriques"

1.3 Règles de formalisation d'une métrique

Au cours du chapitre 1 § 3.1.2, nous avons défini une métrique comme « une expression mathématique permettant d'évaluer la conception d'un système selon un objectif défini et ce, en identifiant des indicateurs (représentant les contraintes de la conception) caractérisés par un ou plusieurs paramètres influents ». Par conséquent, une métrique peut être la somme de plusieurs indicateurs. Toutefois, sa valeur restera sans unité dans un intervalle borné. Chaque borne correspondant à l'atteinte de l'objectif idéal ou à éviter.

Pour qu'une métrique soit efficiente, à l'intérieur d'une entreprise multi-site ou entre plusieurs services, et permettre la comparaison de plusieurs générations de produits, elle doit respecter un certain nombre de règles, telles que nous avons détaillées dans le chapitre 1 §3.2.1.

La valeur de la métrique doit être normée pour garantir que ses valeurs soient comparables les unes aux autres. Les métriques sont ainsi souvent un rapport avec des valeurs comprises entre 0 et 1.

Ensuite, une échelle d'intervalles doit être choisie afin que la différence entre l'évaluation de deux éléments distincts reste significative. Ce choix doit être fait avec les experts, les architectes systèmes et les différents concepteurs intervenant dans la phase de préconception afin que ce soit transparent pour chacun. Pour cela, on peut s'appuyer sur des travaux existants portant sur l'interprétation sémantique par différents individus (Arduin et al. 2013).

De plus, les métriques doivent être compréhensibles et interprétables de la même façon par tous les utilisateurs.

Pour finir, afin de rendre les métriques utilisables, leurs paramètres d'entrée doivent être déterminables: soit physiquement par des tests, soit par simulation ou soit par retours d'expérience. Bien sûr, les méthodes de mesures correspondantes doivent être également définies au préalable.

2 Description du processus

2.1 L'analyse des besoins

Les parties prenantes doivent définir leur besoin relatif à la métrique à concevoir. Par ailleurs, ce besoin peut être très différent d'une entreprise à l'autre et notamment dépendre du moment où la métrique sera utilisée. Habituellement, avant que le système ne soit réalisé physiquement, les parties prenantes sont généralement des membres opérationnels de l'entreprise tels que les responsables, les ingénieurs systèmes, les architectes système, les concepteurs. Enfin, lorsque le produit mécatronique est fabriqué, les acteurs deviennent des clients ou des commerciaux.

Les attentes et / ou les besoins des parties prenantes doivent être la clé de cette approche de sorte que les nouvelles métriques créées pour « la communauté mécatronique » répondent réellement à la spécificité de la conception mécatronique.

Ces acteurs sont susceptibles de définir librement un objectif de conception, soit d'en choisir un dans une bibliothèque d'objectifs prédéfinis. Naturellement, cet objectif pourra être relatif à quelques indicateurs déjà prédéfinis dans leur propre processus industriel.

2.2 Les objectifs à évaluer

Une fois que les besoins industriels ont été analysés, un ou plusieurs « objectifs de conception » doivent être définis pour être évalués par des métriques. Suivant leur type, ces métriques peuvent être classées en 4 catégories :

- métrique de produit,
- métrique de processus,
- métrique de projet (Kan 2002b),
- métrique de système (à concevoir).

Les métriques de produit évaluent les caractéristiques du produit réalisé tel que la fiabilité, les performances (Bourne et al. 2003; Nudurupati et al. 2011).

Les métriques de processus évaluent entre autre, la qualité, les aspects collaboratifs, la complexité, etc. d'un processus (par exemple la conception, l'installation, la maintenance, le processus d'élimination). Elles sont généralement basées sur des considérations de temps, mais aussi sur les boucles, les modes de retour (Kasser and Schermerhorn 1994).

Les métriques projet décrivent les caractéristiques d'un projet et son mode d'exécution, à travers le nombre de participants et leur niveau de compétences, les considérations de calendrier, les coûts relatifs, la structure organisationnelle (de Haas and Kleingeld 1999).

Pour finir, *les métriques de système* sont liées à l'évaluation du système à être conçu avant qu'il ne devienne un produit physique. Elles peuvent par exemple servir à mesurer la qualité des modèles de données, comme le précise Moody (Moody 1998) qui en définit 8 objectifs (l'intégrité, la flexibilité, la compréhension, l'intégration, etc.). Elles aident, pendant l'étape de conception, à la prise de décision liée au choix des architectures, les objectifs à évaluer sont alors la fiabilité, la sécurité, le coût, la robustesse, etc. d'un système, dépendant des choix de conception pris.

2.2.1 Les objectifs de conception

Nous donnons de manière non exhaustive, sur La Table 4-1 une liste des principaux objectifs de conception identifiés pour la mécatronique, classés selon les quatre catégories décrites ci-dessus.

Métrique de produit	Métrique de processus	Métrique de projet	Métrique de système
Valeur acquise	Qualité	Coût	Coût
Qualité	Robustesse	Durabilité (développement durable)	Fonctionnalité
Recyclabilité	Durabilité (développement durable)	Sécurité	Intégration
Maintenabilité	Sécurité	Innovation	Modularité
Durabilité	Complexité	Intégration	Compacité
Disponibilité	Collaboration	Robustesse	Multi-domaine
Sécurité	Interfaces multi-domaines	Maturité	Robustesse
Innovation	Planning		Complexité
Intégration			Fiabilité
Modularité			Recyclabilité
Compacité			Durabilité
Robustesse			Sécurité
Fiabilité			
Autonomie			
Accessibilité			
Complexité			
Assemblage/démontrabilité			
Usage			

Table 4-1 : Quelques objectifs pour la mécatronique

2.2.2 Les intérêts de ces objectifs pour la mécatronique

Nous pouvons dresser de manière non exhaustive les intérêts qui peuvent être relevés lors de l'exploitation des métriques pour la conception mécatronique. Il s'agit :

- Du niveau d'innovation,
- De la collaboration et des interfaces multi-domaines,
- De la compacité.

S'agissant de l'évaluation du niveau d'innovation d'un projet mécatronique est cruciale à cause de l'évolution rapide des attentes de notre société. En effet, ces dernières sont basées sur les nouvelles technologies d'information et les rapides développements technologiques d'objets toujours plus intégrés et plus connectés grâce à l'internet des objets (« Internet of things ») et aux systèmes cyber-physiques (Horváth 2012), alors que le temps de mise sur le marché devient de plus en plus rapide (Tomizuka 2002).

En ce qui concerne la *collaboration* et les *interfaces multi-domaines* dans le processus de conception des systèmes mécatroniques sont particulièrement importants (Gausemeier et al. 2009; Hehenberger 2012; Chen Zheng 2014; Hammadi et al. 2014b) du fait de l'intégration de composants multi-domaines, de l'implication des différentes équipes techniques et de la gestion des problèmes d'interfaces correspondants, d'où la nécessité d'un langage commun pour une vue unique des besoins, revue collaborative fréquente,

gestion des paramètres partagés, arbitrages liés au choix de solutions optimales... Ces métriques évaluent ainsi comment le processus gère efficacement les entités hétérogènes. De plus, et compte tenu que la mécatronique est une discipline qui intègre les technologies de l'ingénierie mécanique, électrique/électronique, de l'automatique et de l'informatique et qu'elle fournit des produits, des processus, et des systèmes améliorés, *l'intégration* est également un objectif clé dans le processus de conception des systèmes mécatroniques, que ce soit pour l'intégration fonctionnelle, l'intégration multi-domaine ou l'intégration physique (Isermann 2007; Craig 2009; Warniez et al. 2012).

Concernant la *compacité*, cette dernière est également un objectif essentiel des systèmes mécatroniques. L'augmentation du nombre de composants à intégrer dans un volume compact fournit des couplages multi-physiques qui peuvent être plus ou moins pressentis. Les métriques géométriques, dans la perspective d'évaluer cette compacité du système évaluent souvent le placement des composants (Warniez et al. 2014b).

Dans la littérature, les objectifs les plus étudiés sont la *fiabilité*, la *disponibilité*, la *maintenabilité*, la *sécurité*, la *robustesse*. Cependant ces objectifs ont été évalués indépendamment pour chaque discipline de la mécatronique mais rarement en prenant en compte l'intégration de ces disciplines (Hammouda et al. 2013). Concernant l'évaluation d'un projet, l'objectif le plus courant reste celui du coût.

2.3 Identification des paramètres influents

Les industriels ont généralement accès à de très nombreuses données, que ce soit sur leur produit, leur processus, leur projet ou le système qu'ils conçoivent. Dans un premier temps, il est donc nécessaire de sélectionner celles (paramètres influents évidents) qui vont permettre d'établir la métrique désirée. Ces paramètres peuvent être fixés soit dans une approche « Top down » par déduction logique des exigences, soit par une démarche « Bottom up » issue de l'expérience de l'entreprise. Dans les faits généralement les deux sont utilisées (Chapitre 1 §1.2.2.1), même si la démarche « Bottom-up » augmente le nombre de boucles d'itération (Hehenberger 2012).

Enfin, ces paramètres influents peuvent également être déterminés par la propre expérience du concepteur, pour cela une bonne pratique est d'interviewer les personnes impliquées.

Dans la littérature, selon l'objectif défini, certains paramètres influents usuels peuvent être identifiés. Nous pouvons citer pour :

- Les métriques de coût : les bénéfices, la valeur (Lee and Paredis 2014), les résultats financiers (Smith 2013),
- Les métriques de planning : le temps d'un processus, le temps pour une tâche (étape de conception, simulation, fabrication, assemblage) (Kitchenham and Linkman 1990),
- Les métriques de qualité : qualité d'un produit (performances), d'un processus (normes), qualité de services (QoS) (Yassa and Sublime 2013) et la satisfaction client (qualité perçue).

Néanmoins, un point important est que pour un même objectif fixé, les paramètres influents peuvent être différents selon le domaine ou le niveau de conception impliqué.

A titre d'exemple, pour l'objectif du développement durable et d'écoconception, les paramètres influents pour une métrique de produit dans le domaine du transport sont par exemple la consommation (KWh/Km/passager), le poids/passager, le taux de recyclabilité, le

taux de substances dangereuses, le taux de matériel renouvelable, le niveau de bruit sonore, la concentration d'émission de particules (Andries 2010).

Pour les métriques de système lors de la conception et fabrication, les paramètres d'écoconception sont la masse vierge nécessaire, les kilomètres parcourus par les pièces, les sous-ensembles jusqu'à l'assemblage; l'énergie spécifique des matériaux entrant dans la composition du produit; le volume des emballages; la contribution à des problèmes de pollutions ou de déchets, etc. Il faut aussi prendre en compte le démontage et le démantèlement lorsque le produit n'est plus utilisé ainsi que le recyclage, la valorisation du produit (Scaravetti 2004).

En parallèle, la faisabilité de la collecte des données correspondantes à ces paramètres, dans une base de données, n'est pas à négliger. L'extraction manuelle des paramètres peut prendre un temps considérable, est souvent laborieuse avec un risque d'erreur probable.

Ces paramètres peuvent être des valeurs discrètes ou non, ou prédéfinis dans une plage de valeurs. Pour vérifier si les paramètres choisis sont réellement influents pour la métrique, une analyse de la sensibilité peut être effectuée (§ 2.6).

La connaissance acquise par expérience dans l'entreprise est importante à cette étape. En effet, si un facteur oublié se trouve être un facteur d'influence, la métrique résultante peut avoir des conséquences préjudiciables pour les prises de décision correspondantes.

2.4 Construction de la métrique

Nous venons de voir qu'une métrique permettait d'évaluer un objectif de conception en se basant sur des paramètres influents.

La métrique est une combinaison mathématique d'un ou plusieurs paramètres directement mesurables, qui doit respecter les règles rappelées au § 1.3. Elle peut être construite à partir de fonctions mathématiques de base comme la somme, la soustraction, la multiplication, la division, mais aussi une fonction polynomiale, logarithmique, exponentielle, ou des équations plus complexe. Elle doit intégrer tous les paramètres influents identifiés.

Un autre élément important pour construire cette métrique est qu'elle doit refléter le « ressenti » de l'entreprise. Pour ce faire, le concepteur de la métrique devra utiliser au moins deux scénarios tests de référence. Il choisira les scénarios pour lesquels il connaît les valeurs attendues de la métrique et/ou leur position relative vis-à-vis de l'objectif évalué, et pourra ainsi prévoir « la courbe de tendance » de la fonction mathématique à choisir.

D'autre part, pour faciliter l'usage de cette approche par les industriels, des métriques prédéfinies (*Librairie de métriques*) basées sur la littérature ou sur leurs propres développements, comme par exemple les métriques précédemment établies aux chapitres 2 et 3. Des fonctions mathématiques de base (*Librairie des fonctions mathématiques*), peuvent être également proposées pour aider le concepteur de métriques à construire rapidement sa métrique.

Par ailleurs, la métrique peut être composée de plusieurs indicateurs et plusieurs méthodes d'agrégation permettent de coupler l'ensemble de ces indicateurs, comme cela a été précédemment discuté à la section 5 du Chapitre 2. Enfin, le concepteur peut privilégier un indicateur par rapport à un autre (Chapitre 1 § 3.1.2) et réaliser une analyse de la sensibilité (§ 2.6), pour améliorer la finesse/adéquation de la métrique élaborée.

2.5 Validation de la métrique

Un élément souvent manquant dans les métriques proposées dans la littérature est leur validation expérimentale. Or, il est fondamental que le concepteur de la métrique compare le résultat de sa métrique sur plusieurs autres scénarios industriels afin de s'assurer que la valeur de la métrique alors calculée est réaliste.

La principale difficulté de l'évaluation sur des cas réels réside alors dans la nécessité de se préoccuper au préalable de la collecte des données. En effet, la collecte des données doit fournir des informations utiles pour l'évaluation du processus, du projet et du produit, sans devenir un fardeau pour les équipes de développement.

Par conséquent, il est important, dans la mesure du possible, de définir précisément les données nécessaires pour la métrique considérée, afin d'éviter un gaspillage de données collectées excessivement.

Néanmoins, au début du processus de construction de la métrique, et ce avant de l'avoir définie, nous avons besoin de collecter beaucoup de valeurs de paramètres sur les scénarios de référence, pour identifier précisément les informations pertinentes et donc les paramètres influents et éventuellement pouvoir faire une analyse de sensibilité.

Enfin, dans certains cas, ces données ont besoin d'être validées avant d'être utilisées (Kan 2002b), notamment lorsqu'une routine de validation n'a pas été implémentée dans la collecte automatique de données.

Cette étape de validation est cruciale dans le processus, car si une différence importante apparaît entre ce que les métriques créées évaluent et ce que les managers ou les concepteurs attendent ou expérimentent, le processus de l'usine à métriques serait un échec.

Il est donc important que les métriques produites soient validées sur des projets réels avec des produits réels et une équipe de développement réelle.

Enfin, dans la mesure où les résultats relatifs aux nouveaux scénarios ne satisfont pas les attentes des industriels, alors une nouvelle analyse de sensibilité des paramètres peut être réalisée pour affiner/améliorer le choix des paramètres influents et la formule mathématique de la métrique, jusqu'à ce qu'elle soit correcte.

Dans les faits, cette validation peut prendre du temps et être réalisée en deux étapes :

- Lors de la construction de la métrique pour la première fois, la validation se fait sur deux ou trois cas scénarios actuels en se basant sur le « ressenti » du concepteur concernant les valeurs de métriques correspondantes.
- Puis, pour une validation plus exhaustive, qui permettra aux responsables de prendre des décisions, la métrique devra être testée sur plusieurs années, plusieurs systèmes, plusieurs processus, plusieurs produits et éventuellement ajustée en fonction des retours.

2.6 Analyse de la sensibilité

Afin de faciliter les étapes précédentes (identification des paramètres influents (§2.3) et construction de la métrique (§2.4)), une analyse de la sensibilité peut être réalisée.

En effet, l'analyse de sensibilité permet d'analyser un modèle en étudiant l'impact de la variabilité des facteurs d'entrée du modèle sur la variable de sortie.

L'idée, ici, est de fournir par exemple par un affichage graphique, la variation des valeurs de la métrique étudiée en fonction de chaque paramètre influent, en vue de vérifier que son évolution en fonction de chacun de ces paramètres corresponde à la valeur attendue de la métrique par le concepteur. A ce titre, plusieurs méthodes existent pour effectuer cette analyse.

Une première méthode est de faire varier les paramètres d'entrées et de regarder les changements à la sortie. Dans (Guo and Gershenson 2003), l'analyse de la sensibilité a été abordée par la variation des entrées puis en mesurant le pourcentage de changement en sortie par rapport à la valeur de la sortie originale. Les variations des entrées sont de 1% et 0,1%. Ces valeurs produisent une image fidèle de la sensibilité aux variations fines.

Par ailleurs, les plans d'expérience permettent, entre autres, de déterminer les facteurs clés dans la conception d'un nouveau produit ou d'un processus (Tinsson 2010). Dans notre cas, l'étude a pour objectif d'identifier les paramètres influents. Selon la méthode traditionnelle, un paramètre est fixe au centre du domaine et un deuxième varie aux extrémités du domaine et vice-et-versa. Avec la méthode des plans d'expérience, les essais sont réalisés aux extrémités du domaine expérimental, la précision obtenue est donc supérieure. Néanmoins, il faut trouver les « bonnes » expériences à réaliser.

La méthode Taguchi, inventée par Gen'ichi Taguchi, est une méthode statistique pour réaliser des plans d'expérience. Appliquée dans l'industrie, cette méthode est principalement utilisée pour améliorer la qualité des produits en limitant les variations autour d'une valeur de consigne. L'objectif général étant d'obtenir dans la mesure du possible des produits, des processus et des systèmes aussi robustes et insensibles aux perturbations externes. Pour ce faire, Taguchi réalise une analyse de sensibilité basée sur les plans d'expériences, dont le but est de simplifier le protocole expérimental et ce, afin de mettre en évidence les effets de facteurs sur la réponse.

Ainsi, la méthode de Taguchi se distingue par une réduction importante du nombre d'essais, tout en gardant une bonne précision. Elle place le modèle comme un élément clé de la stratégie du plan d'expériences. L'expérimentateur choisit librement les facteurs et les interactions à étudier selon le modèle qu'il propose, en étroite adéquation avec ses objectifs. Néanmoins, la place de chaque facteur dans le plan de Taguchi est choisie selon la difficulté de réalisation du facteur dans l'expérience, pour limiter le changement de valeurs possibles dans les paramètres les plus difficiles à faire varier (Sabre 2007).

Enfin, l'analyse de sensibilité peut être menée à l'aide des nombreux outils commerciaux implémentant la méthode six sigmas. A l'origine, cette approche a été initialement développée par Motorola en 1986 pour l'amélioration de la qualité du processus de fabrication (Billam and Pathy 2002). Elle se base sur une démarche fondée à la fois sur la voix du client et sur des données mesurables et fiables.

Cette méthode s'inscrit dans une démarche de réduction de la variabilité dans les processus de production et des produits, conduisant au développement de nombreux outils logiciels d'analyse statistique implémentant l'étude de la variabilité des paramètres. Nous pouvons citer : GNU R⁵⁰, Statgraphics, Statistica ...

⁵⁰ <http://www.r-project.org/>

3 Application de la méthode

Pour illustrer le processus proposé précédemment, nous avons choisi d'adresser l'objectif de « modularité » qui est particulièrement sensible pour les systèmes mécatroniques intégrés, puisque généralement modularité et intégration sont des objectifs contraires dans le choix d'architectures.

3.1 L'analyse du besoin

Si jusqu'à présent le principal enjeu des systèmes mécatroniques était l'intégration, actuellement, la conception modulaire est devenue une stratégie de développement largement acceptée dans de nombreuses industries, même mécatroniques.

Le but visé étant de créer des produits et des systèmes qui peuvent être facilement fabriqués, reconçus et maintenus par différentes équipes, différentes méthodes de fabrication etc., en utilisant des modules prédéfinis comme base de développement des produits et de ses variantes (Holttä-Otto and de Weck 2007).

En effet, la modularité basée sur la décomposition d'un système en sous-systèmes et composants, facilite leur standardisation et augmente la variété des produits (Guo and Gershenson 2003).

Ulrich et al. (Ulrich and Eppinger 2011) confirme cet intérêt en précisant que « la caractéristique la plus importante d'une architecture produit est la modularité ».

Les entreprises s'efforcent désormais de rationaliser leurs lignes de produits en fournissant une augmentation de la diversité des produits à bas coût, renforçant ainsi l'objectif de modularité. En effet, la modularité d'un produit a une forte influence sur son coût initial, sa facilité d'entretien (désassemblage et réassemblage), sa facilité de mise à jour, de conception et de test, le délai de mise en œuvre (Newcomb et al. 1998; Gershenson et al. 2003).

Par ailleurs, et compte tenu que les produits doivent être facilement démontables et permettre rapidement le tri des différents matériaux pour un meilleur recyclage des matériaux et de leurs composants, la modularité présente l'avantage d'un démontage et d'un démantèlement plus faciles. Cela est particulièrement vrai pour les systèmes mécatroniques qui incluent de nombreux composants multi-domaines.

Bien que la modularité ait été de plus en plus appliquée dans la conception de produits industriels au cours de ces deux dernières décennies, il existe peu de moyens pour évaluer cette modularité en répondant aux attentes spécifiques de chaque entreprise.

3.2 Définition de l'objectif

La modularité peut être traitée suivant différentes vues :

- la modularité d'un processus,
- la modularité d'une ligne de production (Siddique et al. 1998),
- la modularité des équipes,
- la modularité des composants/sous-systèmes, etc. (chapitre 1 § 4.2.1).

Suivant le type de modularité, des objectifs différents peuvent être définis. Ici, l'objectif choisi est d'adresser la modularité d'un système, au travers de la modularité des architectures pendant la phase de conception.

3.3 Identification des paramètres influents

Par analogie avec l'évaluation de la modularité des architectures dans le domaine du logiciel proposée par Sant'Anna (Sant'Anna et al. 2007), tout en considérant un module comme un groupement physique ou conceptuel de composants d'un domaine, les paramètres influents pourraient être :

- le nombre de flux d'entrée,
- le nombre de flux de sortie de chaque module, et ce quel que soit la nature des flux présents (informationnels, d'énergie, de matériaux, de position) (Pimmler and Eppinger 1994).

Il pourra être intéressant au contraire de dénombrer les interactions entre composants suivant leur nature (Pimmler and Eppinger 1994) pour étendre la métrique de modularité à différents points de vue :

- Le positionnement : la nécessité d'adjacence ou d'orientation entre les éléments.
- L'énergie : le besoin de transfert d'énergie entre deux éléments.
- L'information : le besoin en information ou transfert de signal entre deux éléments.
- Le matériau : le besoin en échange en matière entre deux éléments.

Selon l'architecture adressée (fonctionnelle, logique ou physique), d'autres paramètres pourront être ajoutés. Dès lors, les architectures fonctionnelles feront apparaître le nombre de fonctions, et associées aux architectures physiques, un paramètre important pourra être de quantifier le nombre de fonctions intégrées par composant par exemple.

3.4 Construction de la métrique

Compte tenu que nous n'avons pas eu la possibilité de récupérer de données sur un scénario réel pour construire la métrique de modularité, nous avons donc choisi quelques exemples de métriques existantes issues de la littérature.

3.4.1 Hehenberger & al. (2005) : modularité fonctionnelle

Une première métrique proposée par Hehenberger (Hehenberger and Zeman 2005) concerne la modularité fonctionnelle. Pour connaître si un produit est fonctionnellement modulaire, la métrique s'écrit (en supposant que le nombre de modules soit inférieur ou égal au nombre de fonctions, lui-même supérieur à 1) :

$$Modularité_1 = \frac{\text{Nombre de modules} - 1}{\text{Nombre de fonctions} - 1} \quad (3.15)$$

Cette métrique établit une relation entre le nombre de fonctions et le nombre de modules qui sont nécessaires pour la réalisation de toutes les fonctions. Elle est normalisée entre 0 et 1. Lorsqu'il y a un maximum de solutions modulaires (le nombre de modules est égal au nombre de fonctions), la valeur de cet indicateur est maximale et est égale à 1. A l'inverse, lorsque la valeur vaut 0, la solution est entièrement intégrée, puisqu'il n'y a qu'un seul module.

Si le nombre de fonctions est inférieur au nombre de modules, la fonction doit être décomposée en plusieurs sous-fonctions. Cela rend cette métrique très subjective,

puisqu'elle est dépendante du niveau de décomposition fonctionnelle choisi par chaque concepteur.

3.4.2 Zhang et al. (2003) : modularité pour le cycle de vie (et coût)

Pour Zhang et al. (Zhang and Gershenson 2003), la modularité idéale d'un module est atteinte lorsque chaque composant du module est indépendant de tout autre composant non contenu dans le module. L'objectif est donc de transformer l'architecture par élimination de modules pour réduire les coûts pendant les étapes du cycle de vie.

La mesure de la modularité relative totale (TRM) est alors définie comme la somme du ratio entre les similitudes intra-modules (S_{in}) et toutes les similitudes intra- et inter-modules ($S_{in}+S_{out}$), et du ratio entre les dépendances intra-modules (D_{in}) et les dépendances totales intra- et inter-modules ($D_{in} + D_{out}$).

$$TRM = \frac{\sum S_{in}}{\sum S_{in} + \sum S_{out}} + \frac{\sum D_{in}}{\sum D_{in} + \sum D_{out}} \quad (3.16)$$

Cette métrique prend en compte les deux aspects classiques de la modularité sur la « similitude » et le « couplage » proposés par Ulrich et Tung cités dans (Gershenson et al. 2003) et repris par Hölta-Otto (Holta-Otto and de Weck 2007). Elle n'est pas normalisée et donc est plutôt adaptée à l'évaluation de l'avancement du processus de reconfiguration modulaire, entre deux architectures successives pour quantifier l'effort de modularisation.

3.4.3 Newcomb & al. (1998) : modularité de conception pour le cycle de vie

Une dernière métrique est proposée par Newcomb & al. (Newcomb et al. 1998) complète l'approche classique de la modularité décrite précédemment en prenant en compte la modularité des architectures d'un même produit composé de module, évalués suivant des points de vue différents du cycle de vie. Elle est composée de deux indicateurs : l'indicateur de correspondance (CR) et l'indicateur d'interactions (CI)

$$Modularité_2 = 0,5 \times (CR_{overall} + CI) \quad (3.17)$$

L'indicateur du rapport de correspondance(CR)

L'indicateur du rapport de correspondance (CR) est une mesure de la correspondance de deux classes d'équivalence (d'un point de vue donné) (V_i et V_j) d'un composant x appartenant à un module donné par rapport au nombre total d'éléments contenus dans ce module.

$$CR = \frac{|V_i(x) \cap V_j(x)|}{|V_i(x) \cup V_j(x)|} \quad (3.18)$$

x représente un composant dans un produit ou un sous-assemblage.

$V_i(x)$ désigne le module d'un point de vue V_i , qui contient le composant x . Les points de vue sont par exemple les matériaux compatibles, les caractéristiques de recyclage des matériaux, la phase « après-vie » (broyé, incinéré, recyclé, réutilisé), la fréquence de service (jamais, rarement, fréquemment), la fonctionnalité, la structure, ...

L'équation (3.18) compare la modularité d'un module considéré dans suivant deux points de vue différents. Pour mesurer la correspondance multi-vue des modules pour un produit entier, il faut évaluer la moyenne des CR de tous les modules dans le produit :

$$CR_{overall} = \frac{\sum CR_i}{nbre\ de\ Modules} \quad (3.19)$$

Les valeurs de cet indicateur sont comprises entre 0 et 1. Lorsqu'il vaut 0, cela indique qu'il n'y a aucune correspondance entre les modules selon les différents points de vue alors que lorsqu'il se rapproche de 1, la correspondance des modules du produit est forte, quel que soit le point de vue considéré.

L'indicateur d'interactions(CI)

Afin d'adresser la seconde propriété de la modularité qui est de minimiser les interactions auxiliaires, un indicateur d'interactions (CI) a été développé. Les dépendances entre les modules peuvent être représentées par les connexions physiques existantes entre eux.

Prenons par exemple, les connexions entre et dans les modules sont intéressantes à connaître, notamment pour déterminer l'effort de séparation des composants pour le recyclage, la réutilisation.

Par conséquent, le second indicateur d'indépendance est fonction du rapport entre les connexions internes au module et la totalité des connexions dans le produit.

$$CI = \frac{connection\ interne\ du\ module}{total\ des\ connexions\ du\ produit} \quad (3.20)$$

La valeur maximale de cet indicateur est 1. Cette valeur signifie que le produit est un ensemble disjoint de modules, qui dans ce cas pourraient être considérés comme des produits séparés.

La valeur minimale de cet indicateur est 0. Elle indique que chaque module constitue un seul et unique composant. Cet indicateur relève ainsi l'autonomie d'un module : plus sa valeur se rapproche de 1 et plus le module est autonome.

3.4.4 Turki (2008) : Métrique de modularité des blocks

Turki (Turki 2008a) a également créé une métrique sur l'autonomie des blocks dans une architecture. Il définit les connections strictement interne par des connections ne faisant pas intervenir de connecteurs à la frontière du module. Ensuite, il définit l'indicateur d'autonomie comme le rapport entre le nombre de liens strictement internes divisés par le nombre de tous les liens que contient le module.

$$Indicateur\ d'autonomie = \frac{Nombre\ de\ liens\ internes}{Nombre\ de\ liens\ totaux} \quad (3.21)$$

Cela signifie qu'il s'intéresse au rapport entre les échanges internes au module et les échanges vers l'extérieur. Plus le résultat de cet indicateur est proche de 1, plus le module est autonome et par conséquent la réutilisation est améliorée.

3.5 Validation de la métrique

Cette étape de validation est cruciale, en effet, si une distorsion importante existe entre la valeur de l'évaluation des métriques construites et celle que les décideurs ou les concepteurs attendent ou expérimentent, le processus de l'usine métrique serait un échec. Ainsi, les métriques conçues doivent être validées sur des projets réels avec de vrais produits et par du personnel de développement réel. Les mesures existantes trouvées dans la littérature manquent souvent de validation expérimentale. Cependant les concepteurs ont besoin de comparer les résultats de la métrique construite sur plusieurs autres cas de test, pour voir comment cette mesure reste valable dans leur environnement industriel. Si les valeurs relatives à chaque cas de test ne répondent pas à leurs attentes, une nouvelle analyse de sensibilité des paramètres peut être effectuée pour affiner ou améliorer le choix des paramètres influents et de la forme mathématique de la métrique, jusqu'à ce qu'une métrique adéquate soit définie.

En fait, cette étape de validation peut prendre du temps et être réalisée en deux phases. Lors de la construction initiale de la métrique, la validation est effectuée sur deux ou trois cas-tests communs, en respectant "le ressenti" des concepteurs sur les valeurs de chaque cas-test. En réalité, pour fournir une mesure utile pour faciliter la prise de décisions, la métrique doit être testée pendant plusieurs années, sur divers systèmes, processus, produits, et peut-être être ajustée en fonction de leurs évaluations, pour être définitivement validée.

Cette validation est d'autant plus importante, lorsque les données sont incertaines ou non matures. Dans ces cas-là, il serait intéressant dans de futurs travaux, d'y introduire ces éléments au travers par exemple de réseaux bayésiens (Moullec et al. 2012a).

Par ailleurs, il faut s'assurer que la métrique respecte les règles énoncées au chapitre 1 §, notamment pour la pertinence de l'échelle des intervalles, la compréhensibilité et la faisabilité.

Guo et al. (Guo and Gershenson 2004) ont comparé différentes métriques de modularité sur quatre produits. Selon Guo, la méthode de conception d'un produit modulaire de Zhang et al. (Zhang and Gershenson 2003) est la méthode la plus fiable. En effet, sur les quatre cas tests, cette méthode obtient les meilleurs résultats en fonction de l'objectif fixé du concepteur.

4 Discussion et perspectives

Un point important lié à ce processus d'usine à métriques est la question de la collecte des données. En effet pour construire une métrique efficiente, il est essentiel de pouvoir sélectionner objectivement les paramètres (donc les données nécessaires) qui serviront à calculer la métrique. Lorsque les données empiriques sont correctement collectées, la prévision des paramètres influents et la forme mathématique adéquate de la métrique sont plus faciles.

De plus, le processus propose la création d'une métrique spécifique à un objectif d'une entreprise. Or généralement, différents objectifs doivent être pris en considération pour faciliter les compromis entre différents points de vue (disciplines, concepteurs, services).

Dans ce cas, le processus de construction d'une métrique peut être utilisé pour chacun de ces objectifs, puis de nombreuses techniques existantes d'optimisation multi-objectifs (Engau 2009; Armaghan and Renaud 2012; Biltgen and Mavris) ou de pondération permettront de coupler ces différentes métriques pour atteindre une métrique globale adéquate.

Concernant le déterminisme de la métrique, la garantie de la pertinence et l'évaluation réaliste de la métrique créée se base sur l'utilisation de métriques normalisées et de cas d'utilisation réels pour la validation.

Si une métrique doit être utilisée sur une longue période (plusieurs décennies, ou différentes générations de produits), les paramètres influents ou les valeurs de référence pourront évoluer. Dès lors, il devient alors indispensable de reconcevoir la métrique avec une nouvelle calibration. Cet exercice est particulièrement difficile pour les métriques d'innovation ou les métriques de processus, car un changement significatif peut remettre en question la pertinence même de la métrique.

En effet, un produit, même dans la même gamme de produit, peut être complètement différent et son évaluation avec les mêmes paramètres, impossible. A ce titre, examinons plus précisément l'exemple des disjoncteurs. A la différence des premiers disjoncteurs qui avaient pour fonction principale de protéger automatiquement un circuit électrique contre les dommages causés par une surtension ou un court-circuit, les disjoncteurs actuels, eux sont intelligemment connectés et complètement intégrés. Ce qui permet de contrôler la consommation d'énergie, la régulation de l'énergie,..., pour rendre les maisons intelligentes.

Il est probant que la modularité ou l'intégration ne peuvent être considérés avec les mêmes métriques. Avec les progrès de la mécatronique, les attentes des clients et les défis techniques évoluent incontestablement très rapidement. Ce phénomène sera accentué fortement avec la révolution des techniques de l'information (IT) pour les nouveaux systèmes cyber physiques d'autant que les paramètres des métriques basés sur les compétences humaines, les méthodes, les outils actuels ne pourront pas rester les mêmes. Inévitablement, ils ne seront plus adaptés pour prendre en considération les changements de l'environnement économique et sociétal ainsi que les évolutions technologiques.

Par conséquent, il serait peut-être raisonnable de considérer qu'une métrique a une durée de vie, qui détermine la période de temps durant laquelle nous supposons qu'elle restera valide.

Par ailleurs, une caractéristique de base d'une métrique est d'être mesurable. En conséquence, il est important de parvenir à la formalisation d'une expression mathématique intégrant tous les paramètres, et ce même les paramètres « humains » (les compétences humaines, le comportement individuel, les méthodes de travail personnel,...). C'est pourquoi un grand nombre de travaux de recherche actuels se concentrent sur la modélisation de l'« humain », soit par des approches biomimétiques, soit par des études sur le comportement cérébral, dans le but de rendre les systèmes plus « humains » et plus intelligents.

Aussi, pourquoi ne pas imaginer des « métriques intelligentes » utilisant les techniques de l'intelligence artificielle pour être automatiquement enrichies avec de nouveaux paramètres, lorsque la matrice de sensibilité des paramètres change ou lorsque les facteurs humains ne sont plus négligeables dans les résultats obtenus.

Toutefois, pour ce faire, nous aurions besoin au préalable d'automatiser le processus de l'usine à métrique, au travers d'une « plate-forme à métriques », connectée à de nombreuses bases de données ou mieux à un outil PLM...

Finalement, après avoir défini des métriques adaptées et pertinentes pour l'entreprise, nous pourrions imaginer utiliser des outils et techniques de l'intelligence artificielle pour sélectionner dans un ensemble de solutions défini par le concepteur du système, le meilleur concept, la meilleure architecture, les meilleurs composants, etc., à partir des valeurs les plus basses ou les plus hautes des métriques

5 Conclusion

Du fait de l'évolution très rapide des nouvelles technologies et de la réduction du temps de mise sur le marché, les entreprises mécatroniques ont besoin pour rester compétitives d'un moyen pour construire des métriques appropriées.

Ce développement s'inscrit dans la volonté de fournir des informations prédictives au plus tôt dans le cycle de conception, et ainsi réduire les changements de conception tardifs et leurs coûts correspondants exorbitants.

Contrairement à d'autres domaines scientifiques plus matures, il apparaît difficilement imaginable de disposer d'un ensemble « universel » de métriques dans l'ingénierie mécatronique qui puisse être approprié à tous les environnements applicatifs de cette discipline.

Le processus introduit et proposé dans ce chapitre en vue de construire des métriques personnalisées exploitables dès les premières étapes du processus de conception mécatronique permettra une meilleure anticipation des phases suivantes, et une évaluation plus efficace de la qualité du produit.

Les étapes clés de ce processus sont premièrement de définir clairement les besoins relatifs à la métrique à concevoir afin de préciser l'objectif à évaluer. Pour construire cette métrique, il est nécessaire de sélectionner les différentes données de manière à identifier les paramètres influents. Une fois la métrique construite, les dernières étapes sont la vérification et la validation de cette métrique sur des cas tests réels.

Cette méthode a l'avantage de créer des métriques personnalisées facilement. Le point le plus critique est la sélection des paramètres influents si le concepteur n'a pas assez d'expérience. Une analyse de la sensibilité peut faciliter cette étape.

De plus, la méthode a été appliquée sur l'objectif de modularité qui est le pendant de l'intégration. Malheureusement, sans données industrielles suffisantes, nous avons relevé les métriques existantes dans la littérature. Dans l'intention de tester cette méthode plus rigoureusement, des cas tests réels et industriels sont indispensables.

CONCLUSION GENERALE

Constat

Evaluation d'architectures

En partant de l'expression d'un besoin, la préconception consiste d'une part à l'analyser, afin de rechercher les architectures candidates, les évaluer pour permettre de sélectionner la/les architecture(s) du concept à développer qui répond au mieux au besoin initial. Chaque étape du processus de conception permet de spécifier et de raffiner le système à concevoir, afin que sa description soit de plus en plus précise et détaillée. Durant chaque étape, les différents acteurs sont amenés à faire une succession de choix, pour déterminer une solution optimale. Or, pendant la phase de préconception, l'architecte système, en charge de la validation des architectures successives (fonctionnelle, logique et physique), est soumis à des choix délicats, car en début de conception, soit il dispose de peu d'éléments pour comparer les différentes architectures qui s'offrent à lui, soit il doit les confronter à de multiples exigences que le système doit satisfaire. Généralement, il fonde sa prise de décision sur son expérience et n'a pas l'assurance que son choix détermine l'architecture la plus performante au regard des objectifs de conception, surtout dans le cas de nouveaux projets.

Conception des systèmes mécatroniques

La conception d'un système mécatronique met en œuvre plusieurs métiers concevant et réalisant des constituants de domaine, nature et contraintes différents qu'il faut assembler de façon optimale. En effet, l'intérêt des systèmes mécatroniques est que leurs propriétés ne résultent pas d'une simple addition des seules propriétés de leurs composants, mais surtout de leurs interactions. Ainsi, un des enjeux clé de ces systèmes tourne autour du concept d'intégration : le concepteur mécatronicien doit habituellement traiter simultanément les trois objectifs d'intégration suivants : fonctionnelle, multi-domaine et physique.

Besoin

Les produits mécatroniques sont difficiles et longs à concevoir avec des risques de perte de fiabilité, de sécurité, de performances de par la complexité des différents niveaux d'intégration précédents. En effet, la conception de systèmes mécatroniques génère, à partir de spécifications multidisciplinaires, un espace de solutions possibles si large, que les concepteurs ont besoin de moyens efficaces pour trouver la solution optimale qui satisfera les nombreuses contraintes mécatroniques.

Ainsi, même si de nombreux cycles de conception, de méthodes et outils de modélisation existent, l'évaluation des différentes architectures candidates d'un système mécatronique reste un vrai challenge. En effet, en phase de préconception, l'architecte système doit trouver la ou les « bonnes » architectures qui satisfassent les objectifs de conception, et notamment ceux d'intégration des systèmes mécatroniques. Aujourd'hui, les moyens existants à sa disposition ne sont pas suffisants, soit parce qu'ils adressent un domaine en

particulier, ou ne visent pas directement l'objectif d'intégration, primordial pour ces systèmes.

Contributions

Un des moyens permettant d'évaluer quantitativement plusieurs architectures pour faciliter la prise de décision de l'architecte système, tout en justifiant ses choix, et ce dès les phases amont de la conception, est d'utiliser des métriques.

En effet, de manière générale, les métriques sont destinées à mesurer quantitativement la conception des systèmes suivant un ou plusieurs objectif(s) précis. Elles permettent d'augmenter l'efficacité, la productivité et la qualité de la conception des systèmes, tout en limitant les itérations dans la phase de préconception.

Afin d'aider l'architecte système dans la préconception des systèmes mécatroniques, nous avons choisi dans ces travaux de thèse, après avoir fait un état de l'art et contesté des manques pour les systèmes spécifiquement mécatroniques, de développer des métriques pour le choix d'architectures des systèmes mécatroniques en phase de préconception, et plus particulièrement celles pour l'évaluation quantitative de leur degré d'intégration multi-domaine et physique.

Nous nous sommes intéressés tout d'abord aux métriques d'architectures pour l'intégration *multi-niveau et multi-domaine*. Pour cela, nous avons développé *deux métriques*, adressant respectivement ces deux intégrations. Nous avons élargi nos travaux à la prise en compte de l'effort d'innovation dans chacun de ces domaines au sein du système, au travers d'une *métrique d'innovation*. La combinaison de ces trois métriques nous a permis de construire une *métrique de mise en œuvre d'un projet mécatronique*, pour quantifier en amont les efforts financiers, humains et matériels correspondants (le temps total d'heures de travail, le nombre d'équipes techniques associées, les interfaçages nécessaires, le coût du projet,...), et ainsi évaluer la faisabilité du concept.

Afin d'aider le concepteur, nous avons démontré la possibilité d'automatiser le calcul de ces métriques en enrichissant les modèles SysML de stéréotype et en exportant ces modèles vers un fichier de calcul Excel.

Nous avons ensuite abordé l'intégration physique. Pour construire la métrique d'intégration physique, nous avons développé deux indicateurs : un indicateur de compacité et un indicateur d'accessibilité. Le premier évalue l'espace disponible dans le système, et le deuxième évalue la possibilité ou non de rajouter des composants à l'intérieur du système. Pour calculer ces indicateurs, nous avons développé une extension géométrique pour prendre en compte les caractéristiques géométriques des composants en phase de préconception.

De plus, le manque de données exhaustives sur un cas industriel complet ne nous a pas permis de tester l'ensemble des métriques développées sur le même exemple. Cependant, nous avons utilisé deux exemples mécatroniques relativement génériques (le moteur-roue et la motorisation hybride) pour valider nos métriques et montrer la faisabilité de l'automatisation de leurs calculs dans Excel et Catia.

Les métriques que nous avons proposées pour le choix d'architectures en préconception de systèmes mécatroniques adressent principalement l'objectif d'intégration. Or la conception des systèmes mécatroniques nécessitent dans les faits beaucoup plus de métriques, qui sont propres à chaque entreprise. Il nous a donc semblé intéressant de proposer un processus de construction de métriques pouvant adresser n'importe quel objectif de conception. Il permet de donner les définitions et règles de bonnes pratiques pour construire une métrique efficace.

Perspectives

L'un des premiers travaux qui sera mis en œuvre sera d'approfondir « le processus de la fabrique à métriques », dont le concept a été brièvement présenté dans le chapitre 4. En effet, pour approfondir et enrichir cette approche, il restera à la tester sur plusieurs cas industriels, et à l'illustrer avec le développement d'autres métriques dont les objectifs cités dans le chapitre 1 qui sont critiques pour la conception des systèmes mécatroniques, comme la fiabilité, la robustesse, etc.

Par ailleurs, les métriques doivent prendre en compte tous les paramètres influents et ce même les paramètres humains comme les compétences humaines, le comportement individuel ou les méthodes de travail. En se basant sur certains travaux de recherche considérant déjà ces aspects, des métriques intelligentes peuvent être développées avec ces nouveaux paramètres.

De plus, il est important de collecter des données industrielles pour enrichir les bibliothèques et les bibliothèques de notre approche et ainsi créer un démonstrateur, qui pourrait être connecté à un outil PLM.

A terme, comme dans la phase de conception, les réflexions au niveau des architectures se situent surtout au niveau des volumes et de la distribution globale des volumes pour l'ensemble de système, l'objectif serait d'avoir un outil d'aide à la décision pour le choix d'architecture dans la phase de préconception des systèmes mécatroniques, s'appuyant d'une part sur le couplage des indicateurs au niveau système avec les indicateurs géométriques et d'autre part sur le couplage des indicateurs d'intégration multi-domaine et multi-niveau, et des indicateurs géométriques.

Un démonstrateur pourra être développé pour vérifier la faisabilité de son utilisation auprès des industriels. Un partenariat avec une entreprise serait idéal pour avoir un cas industriel mécatronique complet permettant d'appliquer les métriques développées puis d'améliorer et d'approfondir le processus de création des métriques.

Par ailleurs, les métriques d'intégration physique seront enrichies par des métriques de couplages multi-physiques, en collaboration avec les thèses en cours au sein de notre équipe de recherche, pour la validation d'architectures spatiales sous contraintes multi-physiques (thermiques, électromagnétiques, dynamiques ...) en phase de préconception.

Dans le futur, les métriques développées pourront être enrichies et appliquées à d'autres projets plus complets et complexes du laboratoire, avec d'autres applications et secteurs industriels, comme notamment les « Cyber-Physical Systems » ou encore dans le domaine de la production, qui font de plus en plus intervenir des problématiques mécatroniques.

Un dernier point important est la conceptualisation de la définition des métriques. Une piste pour ce travail est par exemple la projection des différents éléments à comparer sur un axe des objectifs à atteindre.

BIBLIOGRAPHIE

- Adamsson N (2005) Management of mechatronics engineering: reflections and propositions. In: Proceedings of 12th International Product Development Management Conference, vol. 1. Copenhagen, pp 35–48
- Addouche S-A, Dafaoui EM, El Mhamedi A (2005) Causality identification of inductor and indicator of performance for process improvement. In: 4th Int. Workshop on Performance Measurement (IFIP2005),. France, p IFIP2005
- Aizier B, Lizy-Destrez S, Seidner C, et al (2012) xFFBD : towards a formal yet simple and complete functional modeling technique for system designers. Rome, Italy,
- Ali MJ (2006) Metrics for Requirements Engineering. Umeå University, UMEÅ, SWEDEN
- Altshuller GS (1984) Creativity as an exact science: the theory of the solution of inventive problems. Gordon and Breach Science Publishers, New York
- Alvarez Cabrera AA, Foeken MJ, Tekin OA, et al (2010) Towards automation of control software: A review of challenges in mechatronic design. *Mechatronics* 20:876–886. doi: 10.1016/j.mechatronics.2010.05.003
- Ameri F, Summers JD, Mocko GM, Porter M (2008) Engineering design complexity: an investigation of methods and measures. *Res Eng Design* 19:161–179. doi: 10.1007/s00163-008-0053-2
- Andries V (2010) Ecodesign in Alstom Transport: Environmental impacts and aspects integration in rolling stock design.
- ANSI/EIA (2003) Processes for Engineering a System. American National Standards Institute
- Arduin P-E, Grundstein M, Rosenthal-Sabroux C (2013) Evaluer la prise en compte des connaissances tacites dans un système d'information. Vers un système d'information et de connaissance. *Ingénierie des systèmes d'information* 18:121–148. doi: 10.3166/isi.18.3.121-148
- Armaghan N, Renaud J (2012) An application of multi-criteria decision aids models for Case-Based Reasoning. *Information Sciences* 210:55–66. doi: 10.1016/j.ins.2012.04.033
- Atlee J, Kirchain R (2006) Operational Sustainability Metrics Assessing Metric Effectiveness in the Context of Electronics-Recycling Systems. *Environmental Science & Technology* 40:4506–4513. doi: 10.1021/es050935l
- Barbedienne R, Penas O, Choley JY, Riviere A (2014) Introduction of Geometrical Constraint Modeling in SysML for Mechatronic Design. Tokyo, Japan,
- Bashir HA, Thomson V (1999) Estimating Design Complexity. *Journal of Engineering Design* 10:247–257. doi: 10.1080/095448299261317

- Baysal MM, Roy U, Sudarsan R, et al (2005) Product Information Exchange Using Open Assembly Model: Issues Related to Representation of Geometric Information. ASME, Orlando, Floride, pp 601 – 612
- Benade SJ, Pretorius L (2012) System architecture and enterprise architecture: a juxtaposition? South African Journal of Industrial Engineering 23:29–46.
- Ben Hamida S, Grandou A, Jankovic M, et al (2015a) A Comparative Case Study of Functional Models to Support System Architecture Design. Jon Wade and Robert Cloutier 44:325–335. doi: 10.1016/j.procs.2015.03.058
- Ben Hamida S, Jankovic M, Callot M, et al (2015b) Towards a Decision Support Framework for System Architecture Design. Milan, Italy, pp 043–052
- Berg DM de, Kreveld DM van, Overmars PDM, Schwarzkopf DOC (2000) Computational Geometry. In: Computational Geometry. Springer Berlin Heidelberg, pp 1–17
- Bettig B, Kale V (2012) Geometric Constraint Solving With Solution Selectors. Journal of Computing and Information Science in Engineering 12:041002.
- Bettig B, Shah J (2001) Derivation of a standard set of geometric constraints for parametric modeling and data exchange. Computer-Aided Design 33:17–33. doi: 10.1016/S0010-4485(00)00058-0
- Bidarra R, Bronsvoot WF (2000) Semantic feature modelling. Computer-Aided Design 32:201–225.
- Billam R, Pathy FV (2002) Amélioration continue dans l'entreprise.
- Biltgen P, Mavris D A Technique for Interactive Probabilistic Multiple Attribute Decision Making. In: 45th AIAA Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics and Astronautics,
- Bishop RH (2007) The mechatronics handbook. CRC Press, The University of Texas at Austin, Texas
- Blanchard BS, Fabrycky WJ (2010) Systems engineering and analysis, 5ème édition. Prentice Hall
- Blessing LTM, Chakrabarti A (2009) DRM, a design research methodology. Springer, Dordrecht ; London
- Blessing LTM, Chakrabarti A, Wallace KM (1998) An Overview of Descriptive Studies in Relation to a General Design Research Methodology. In: Frankenberger D-IE, Birkhofer PH, Badke-Schaub DP (eds) Designers. Springer London, pp 42–56
- Boehm BW (1988) A spiral model of software development and enhancement. Computer 21:61–72.

- Boucher M, Houlihan D (2008) System Design: New Product Development for Mechatronics. Aberdeen Group, Massachusetts, United States
- Bourne M, Neely A, Mills J, Platts K (2003) Implementing performance measurement systems: a literature review. *International Journal of Business Performance Management* 5:1. doi: 10.1504/IJBPM.2003.002097
- Bouyssou D (ed) (2006) Evaluation and decision models with multiple criteria: stepping stones for the analyst. Springer, New York, NY
- Briand L, Morasca S, Basili VR (1998) A goal-driven definition process for product metrics based on properties.
- Briand L, Morasca S, Basili VR (1994) Defining and Validating High-Level Design Metrics. University of Maryland
- Bricogne-Cuignières M (2015) Méthode agile pour la conception collaborative multidisciplinaire de systèmes intégrés : application à la mécatronique. Université de Technologie de Compiègne
- Casner D, Houssin R, Knittel D, Renaud J (2013) Une démarche de conception et d'optimisation de systèmes mécatroniques à partir de l'optimisation multidisciplinaire et basée sur le retour d'expériences. In: 21ème Congrès Français de Mécanique. Bordeaux, France,
- Chen K, Bankston J, Panchal JH, Schaefer D (2009) A framework for integrated design of mechatronic systems. In: Collaborative Design and Planning for Digital Manufacturing. Springer, pp 37–70
- Chen M, Hammami O, Callot M (2014) Architecture framework associated to vehicle architecture definition. In: CESUN 2014. Hoboken, NewYork,
- Chen Zheng MB (2014) Survey on Mechatronic Engineering: A Focus on Design Methods and Product Models. *Advanced Engineering Informatics*. doi: 10.1016/j.aei.2014.05.003
- Choley J-Y (2005) Une approche variationnelle de l'association des références en tolérancement géométrique. Ecole Centrale - SUPMECA
- Choley J-Y, Hammadi M, Roumizadeh B, Penas O (2010) Multi-physics Issues for 3D Electronics. Yokohama, Japan,
- Clarkson KL, Shor PW (1989) Applications of random sampling in computational geometry, II. *Discrete Comput Geom* 4:387–421. doi: 10.1007/BF02187740
- Clarkson KL, Shor PW (1988) Algorithms for Diametral Pairs and Convex Hulls That Are Optimal, Randomized, and Incremental. In: Proceedings of the Fourth Annual Symposium on Computational Geometry. ACM, New York, NY, USA, pp 12–17

- Clément A, Rivière A, Serré P, Valade C (1998) The TTRs : 13 Constraints for Dimensioning and Tolerancing. In: Geometric Design Tolerancing: Theories, Standards and Applications, Hoda A. ElMaraghy. Springer US, pp 122–131
- Clivillé V (2004) Approche systémique et méthode multicritère pour la définition d'un système d'indicateurs de performance. Ecole supérieure d'ingénierie d'Annecy
- Côté V, Bourque P, Oligny S, Rivard N (1988) Software metrics: An overview of recent results. *Journal of Systems and Software* 8:121–131. doi: 10.1016/0164-1212(88)90005-2
- Coulter SL, Bras B, McIntosh MW, Rosen DW (1998) Identification of Limiting Factors for Improving Design Modularity. In: *Proceeding of the 1998 ASME Design Engineering technical Conference*. Atlanta, Georgia, USA,
- Couturier P, Lô M, Imoussaten A, et al (2014) Tracking the consequences of design decisions in mechatronic Systems Engineering. *Mechatronics* 24:763–774. doi: 10.1016/j.mechatronics.2014.03.004
- Craig K (2009) Mechatronic system design. In: *Proceedings of the Motor, Drive & Automation Systems Conference*.
- Crespi V, Galstyan A, Lerman K (2008) Top-down vs bottom-up methodologies in multi-agent system design. *Auton Robot* 24:303–313. doi: 10.1007/s10514-007-9080-5
- Danilovic M, Browning TR (2007) Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Project Management* 25:300–314. doi: 10.1016/j.ijproman.2006.11.003
- Da Silva F, Chevallier G, Saheli M (2012) Model order reduction technique for localized nonlinearities in multiphysics problem - applications in automotive industry. In: *ECCOMAS 2012*. Vienna, Austria,
- Das SK, Yedlarajiah P, Narendra R (2000a) An approach for estimating the end-of-life product disassembly effort and cost. *International Journal of Production Research* 38:657–673. doi: 10.1080/002075400189356
- Das SK, Yedlarajiah P, Narendra R (2000b) An approach for estimating the end-of-life product disassembly effort and cost. *International Journal of Production Research* 38:657–673. doi: 10.1080/002075400189356
- Deepak B (1998) Design to Cost (DTC) approach to Product development. In: *Save International Conference Proceeding*. Save,
- de Haas M, Kleingeld A (1999) Multilevel design of performance measurement systems: enhancing strategic dialogue throughout the organization. *Management Accounting Research* 10:233–261. doi: 10.1006/mare.1998.0098
- Dixon K (2012) Process Integrates Complex 3-D Parts into a Single Device | Qmed. *MPMN* 28:

- DoD D of D (2001) Systems engineering fundamentals. Systems management college
- DSM DSMweb.org: Design Structure Matrix (DSM). In: Massachusetts Institute of Technology (MIT) and the University of Illinois at Urbana-Champaign (UIUC). <http://www.dsmweb.org/>.
- Dubois S (2004) Contribution à la formulation des problèmes en conception de systèmes techniques. Etude basée sur la TRIZ. THESE, INSA de Strasbourg
- Ehrgott M (2005) Multicriteria optimization, 2. ed. Springer, Berlin
- Engau A (2009) Tradeoff-based decomposition and decision-making in multiobjective programming. European Journal of Operational Research 199:883–891. doi: 10.1016/j.ejor.2009.01.061
- Espanet C (1999a) Modélisation et conception optimale de moteurs sans balais à structure inversée - Application au moteur-roue. THESE, Université de Franche-Comté
- Espanet C (1999b) Modélisation et conception optimale de moteurs sans balais à structure inversée - Application au moteur-roue. Université de Franche-Comté
- Espinoza Limón A, Garbajosa Sopeña J (2005) The need for a unifying traceability scheme.
- Estefan JA (2008) Survey of Model-Based Systems Engineering (MBSE) Methodologies. INCOSE
- Faisandier A (2013) Systems Architecture and Design, Engineerign and Architecting Multidisciplinary Ssytems. Sinergy'Com, Belberaud
- Fenstermaker S, George D, Kahng AB, et al (2000) METRICS: a system architecture for design process optimization. In: Proceedings of the 37th Annual Design Automation Conference. ACM, pp 705–710
- Ferguson RW (2004) A project risk metric. The Journal of Defense Software En gineering 16:383.
- Figueira J, Greco S, Ehrgott M (2005) Multiple criteria decision analysis: state of the art surveys. Springer, New York, NY
- FIPS (1993a) FIPS PUB 183, INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0). NIST-FIPS
- FIPS (1993b) FIPS PUB 184, INTEGRATION DEFINITION FOR INFORMATION MODELING (IDEF1X). NIST-FIPS
- Friedenthal S, Moore A, Steiner R (2008) A practical guide to SysML - the systems modeling language. Elsevier ; Morgan Kaufmann, Amsterdam; Boston; Burlington, Mass.

- Fujimoto H, Ahmed A, Sugi K (2001) Product's disassemblability evaluation using information entropy. *IEEE Comput. Soc*, pp 353–359
- Gardan Y (2002) CAO Modelisation Geometrique. Ed. Techniques Ingénieur
- Gausemeier J, Schäfer W, Greenyer J, et al (2009) Management of Cross-Domain Model Consistency during the Development of Advanced Mechatronic Systems. In: ICED 09, Design Methods and Tools (pt. 2) ,. Palo Alto, CA, USA, pp 1–12
- Gershenson JK, Prasad GJ, Zhang Y (2003) Product modularity: Definitions and benefits. *Journal of Engineering Design* 14:295–313. doi: 10.1080/0954482031000091068
- Gillet P (1994) Conception assistée par ordinateur (CAO). Techniques de l'ingénieur
- Grossard M, Boukallel M, Régnier S, Chaillet N (2013) Design of Integrated Flexible Structures for Micromanipulation. In: Grossard M, Chaillet N, Régnier S (eds) *Flexible Robotics*. John Wiley & Sons, Inc., pp 1–35
- Guide VDR, Srivastava R, Kraus ME (1997) Product structure complexity and scheduling of operations in recoverable manufacturing. *International Journal of Production Research* 35:3179–3200. doi: 10.1080/002075497194345
- Guo F, Gershenson JK (2004) A Comparison of Modular Product Design Methods Based on Improvement and Iteration. 261–269. doi: 10.1115/DETC2004-57396
- Guo F, Gershenson JK (2003) Comparison of Modular Measurement Methods Based on Consistency Analysis and Sensitivity Analysis. Chicago, Illinois, USA, pp 393–401
- Gu P, Hashemian M, Sosale S, Rivin E (1997) An Integrated Modular Design Methodology for Life-Cycle Engineering. *CIRP Annals - Manufacturing Technology* 46:71–74. doi: 10.1016/S0007-8506(07)60778-1
- Gupta S, Okudan GE (2008) Computer-aided generation of modularised conceptual designs with assembly and variety considerations. *Journal of Engineering Design* 19:533–551. doi: 10.1080/09544820802527106
- Hammadi M (2012a) Contribution à l'intégration de la modélisation et la simulation multi-physique pour conception des systèmes mécatroniques,. Ecole Centrale Paris, Ecole nationale d'ingénieurs de sfax (Tunisie)
- Hammadi M (2012b) Contribution à l'intégration de la modélisation et la simulation multi-physique pour conception des systèmes mécatroniques,. Ecole Centrale Paris, Ecole nationale d'ingénieurs de sfax (Tunisie)
- Hammadi M, Choley J-Y, Mhenni F (2014a) A multi-agent methodology for multi-level modeling of mechatronic systems. *Advanced Engineering Informatics* 28:208–217. doi: 10.1016/j.aei.2014.03.005

- Hammadi M, Kellner A, Choley JY, Hehenberger P (2014b) Mechatronic design optimization using multi-agent approach. Leo J De Vin and Jorge Solis, Karlstad, Sweden, pp 310–317
- Hammouda N, Habchi G, Barthod C, Duverger O (2013) Mise en oeuvre d'une méthodologie d'évaluation de la fiabilité pour les systèmes mécatroniques. Bordeaux, France,
- Hehenberger P (2012) Advances in Model-Based Mechatronic Design. Trauner Verlag, Linz
- Hehenberger P, Zeman K (2005) Evaluation of Modular Design Concepts of Complex Mechatronic Systems. In: 31st Design Automation Conference. Design Engineering Division and Computers and Information in Engineering Division, Long Beach, California, USA, pp 101–108
- Heininger N, John W, Boßler H-J (2014) Manufacturing of Molded Interconnect Devices from Prototyping to Mass Production with Laser Direct Structuring.
- Heinrich B, Kaiser M, Klier M (2007) How to measure data quality? A metric-based approach.
- Holt J, Perry S (2008) SysML for Systems Engineering. IET
- Holttä-Otto K, de Weck O (2007) Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints. Concurrent Engineering 15:113–126. doi: 10.1177/1063293X07078931
- Horváth I (2012) Beyond advanced mechatronics: new design challenges of Social-Cyber-Physical systems.
- Hsu H-Y, Lin GCI (2002) Quantitative measurement of component accessibility and product assemblability for design for assembly application. Robotics and Computer-Integrated Manufacturing 18:13–27. doi: 10.1016/S0736-5845(01)00020-5
- Hussain A, Ferneley E (2008) Usability Metric for Mobile Application: A Goal Question Metric (GQM) Approach. In: Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services. ACM, New York, NY, USA, pp 567–570
- IEEE (1999a) Standard for application and Management of the Systems Engineering Process.
- IEEE (1998) IEEE SA - 1320.1-1998 : IEEE Standard for Functional Modeling Language - Syntax and Semantics for IDEF0. New York
- IEEE (1999b) IEEE Standard VHDL Analog and Mixed-Signal Extensions. IEEE-SA Standards Board
- IEEE (2000) IEEE Standard VHDL Language Reference Manual. IEEE-SA Standards Board
- Ince DC, Sheppard MJ (1988) System design metrics: a review and perspective. In: Software Engineering. IET, Liverpool, pp 23 – 27

- INCOSE (2006) Systems engineering handbook - A Guide for System Life Cycle Processes and Activities. INCOSE
- Isermann R (2007) Mechatronic systems-Innovative products with embedded control. Control Engineering Practice 16:14–29.
- Ishii K (1998) Modularity: a key concept in product life-cycle engineering.
- Islam A, Hansen HN, Tang PT, Sun J (2009) Process chains for the manufacturing of molded interconnect devices. Int J Adv Manuf Technol 42:831–841. doi: 10.1007/s00170-008-1660-9
- ISO (2001) ISO/CEI 15288 : Systems and software engineering - System life cycle processes. AFNOR
- ISO (2011a) ISO/CEI TR 24748-2 : Systems and software engineering - Life cycle management - Part 2 : guide to the application of ISO/IEC 15288 (System life cycle processes).
- ISO (2002) Software engineering –Product quality – Part 2: External metrics.
- ISO (2014) ISO/IEC/IEEE 42010:2011(E) Systems and software engineering — Architecture description.
- ISO (2011b) ISO 17450-1:2011 - Geometrical product specifications (GPS) -- General concepts -- Part 1: Model for geometrical specification and verification. AFNOR
- ISO (2012) ISO 17450-2:2012 - Geometrical product specifications (GPS) -- General concepts - - Part 2: Basic tenets, specifications, operators, uncertainties and ambiguities. AFNOR
- Ittner CD, Larcker DF, Meyer MW (2003) Subjectivity and the Weighting of Performance Measures: Evidence from a Balanced Scorecard. Social Science Research Network, Rochester, NY
- James L Mathieson JDS (2010) Complexity Metrics for Directional Node-Link System Representations: Theory and Applications. In: Proceedings of the ASME Design Engineering Technical Conference. ASME, Montreal , Quebec , Canada, pp 13–24
- Jankovic M, Bocquet JC, Stal Le Cardinal J, et al (2006) Integral collaborative decision model in order to support project definition phase management. In: DS 36: Proceedings DESIGN 2006, the 9th International Design Conference, Dubrovnik, Croatia.
- Jankovic M, Holley V, Yannou B (2012) Multiple-domain design scorecards: a method for architecture generation and evaluation through interface characterisation. Journal of Engineering Design 23:746–766. doi: 10.1080/09544828.2012.706270
- Jaulin L, Kieffer M, Didrit O, Walter E (2001) Applied Interval Analysis, 2001 edition. Springer, London ; New York

- Jones C (2008) *Applied Software Measurement: Global Analysis of Productivity and Quality*. McGraw Hill Professional
- Kan SH (2002a) *Metrics and Models in Software Quality Engineering*, 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA
- Kan SH (2002b) Software quality metrics overview.
- Kasser JE, Schermerhorn R (1994) Determining metrics for systems engineering. In: *Proceedings of the 4th International Symposium of the NCOSE*.
- Kellner A, Hehenberger P (2014) Hierarchical system modeling for an industrial gantry robot. Karlstad, Sweden,
- Kim K-Y, Manley DG, Yang H (2006) Ontology-based assembly design and information sharing for collaborative product development. *Computer-Aided Design* 38:1233–1250. doi: 10.1016/j.cad.2006.08.004
- Kitchenham BA, Linkman SJ (1990) Design Metrics in Practice. *Information Software Technology* 32:304–310. doi: 10.1016/0950-5849(90)90064-X
- Klee V, al. (1997) *Dictionnaire des mathématiques: algèbre, analyse, géométrie.*, Encyclopaedia Universalis. Albin Michel, Paris
- Klein Meyer J-S, Cabannes G, Khoury I, et al (2007) Optimisation et robustesse pour la prise de décision dans un contexte de conception intégrée et collaborative. Grenoble, France, p 6
- Kramer GA (1991) Using Degrees of Freedom Analysis to Solve Geometric Constraint Systems. In: *Proceedings of the First ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications*. ACM, New York, NY, USA, pp 371–378
- Krebs T, Šegmanović B (2013) Integrating the CAD Worlds of Mechanics and Electronics with NEXTRA. In: *Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment*, Springer. Josip Stjepandić , Georg Rock, Cees Bil, London; New York, p pp 777–788
- Kremer GO, Chiu M-C, Lin C-Y, et al (2012) Application of axiomatic design, TRIZ, and mixed integer programming to develop innovative designs: a locomotive ballast arrangement case study. *The International Journal of Advanced Manufacturing Technology* 61:827–842. doi: 10.1007/s00170-011-3752-1
- Krill JA, Darrin A, Grant CJ, Kedia A (2015) Engineering innovation into systems engineering practices. In: *Systems Conference (SysCon)*. IEEE, Vancouver, BC, pp 27–32
- Lago P, Muccini H, van Vliet H (2008) A scoped approach to traceability management. *Journal of Systems and Software* 82:168–182. doi: 10.1016/j.jss.2008.08.026
- Lawson B (2005) *How Designers Think*, 4th edn. Routledge - Broché

- Lee BD, Paredis CJJ (2014) A Conceptual Framework for Value-Driven Design and Systems Engineering. In: Moroni G, Tolio T (eds) 24th CIRP Design Conference. Milan, Italy,
- Lo M (2013) Contribution à l'évaluation d'architectures en Ingénierie Système : application en conception de systèmes mécatroniques. Université Montpellier II - Sciences et Techniques du Languedoc
- Mani V, Das S, Caudill R (2001) Disassembly complexity and recyclability analysis of new designs from CAD file data. In: Proceedings of the 2001 IEEE International Symposium on Electronics and the Environment, 2001. pp 10–15
- Martin Edin Grimheden (2013) Can agile methods enhance mechatronics design education? *Mechatronics* 23:967–973. doi: 10.1016/j.mechatronics.2013.01.003
- Matthews P, Lomas C, Maropoulos P (2006) A Methodology for Negotiating Change Propagation in Agile Design. Norfolk, VA (USA),
- Matthias Güdemann SK (2010) SysML in digital engineering. 1–8. doi: 10.1145/1837154.1837155
- McAdams DA, Wood KL (2002) A Quantitative Similarity Metric for Design-by-Analogy. *Journal of Mechanical Design* 124:173–182. doi: 10.1115/1.1475317
- Med Amine Ghodhbane (2012) Modèle en cascade.
- Meinadier J-P (1998) Ingénierie et intégration des systèmes. Hermès, Paris
- Message O, Landrieve F (2002) Instrumented ball bearing - SKF, France. SKF, France (Montigny le Bretonneux), United State
- Mhenni F, Choley J-Y, Penas O, et al (2014) A SysML-based methodology for mechatronic systems architectural design. *Advanced Engineering Informatics* 28:218–231. doi: 10.1016/j.aei.2014.03.006
- Mihalache AG (2007) Modélisation et évaluation de la fiabilité des systèmes mécatroniques: application sur système embarqué. Université d'Angers
- Moinet M, Serré P, Rivière A, Clément A (2007) Modelling of the transformation of an object defined by geometric constraints. Rabat, Morocc o,
- Moisan S (2000) Aide à la decision dans le choix de conception. <http://sylvain.moisan.free.fr/publi/ADCC.pdf>.
- Moody DL (1998) Metrics for Evaluating the Quality of Entity Relationship Models. In: Ling T-W, Ram S, Lee ML (eds) *Conceptual Modeling – ER '98*. Springer Berlin Heidelberg, pp 211–225

- Moser D, Krause J (eds) (2006) 3D-MID — Multifunctional Packages for Sensors in Automotive Applications. In: Advanced Microsystems for Automotive Applications 2006. Springer-Verlag, Berlin/Heidelberg, pp 369–375
- Moullec M-L, Bouissou M, Jankovic M, Bocquet J-C (2012a) Product Architecture Generation and Exploration using Bayesian Networks.
- Moullec M-L, Jankovic M, Bouissou M, et al (2013) Proposition of Combined Approach for Architecture Generation Integrating Component Placement Optimization. In: DETC2013-12557. Portland, Oregon, USA,
- Moullec M-L, Jankovic M, Bouissou M, Bocquet J-C (2012b) PRODUCT ARCHITECTURES GENERATION UNDER UNCERTAINTY: COMPARISON BETWEEN TWO METHODS. In: ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. Chicago, IL, USA, p 12
- Moullec ML, Jankovic M, Eckert C (2015) Investigating the Importance of Criteria Selection in System Architecture: Observations from Industrial Experiment.
- Muller G (2011) Systems Architecting: A Business Perspective, Taylor & Francis Group. CRC Press, New York
- National Aeronautics and Space Administration (2007) Systems Engineering Handbook. Washington, D.C.
- Neely A, Mills J, Platts K, et al Performance measurement system design: developing and testing a process-based approach. International Journal of Operations & Production Management 20:1119–1145. doi: 10.1108/01443570010343708
- Newcomb PJ, Bras B, Rosen DW (1998) Implications of modularity on product design for the life cycle. Journal of Mechanical Design 120:483–490. doi: 10.1115/1.2829177
- NF (2008) Mécatronique - Vocabulaire. AFNOR
- NF (1984) Vocabulaire de maintenance et de gestion des biens durables. AFNOR
- NF (1982) Guide pour la prise en compte des critères de maintenabilité des biens durables à usage industriel et professionnel. AFNOR
- NF (2011) NF X50-100 : Management par la valeur - Analyse fonctionnelle, caractéristiques fondamentales - Analyse fonctionnelle : analyse fonctionnelle du besoin (ou externe) et analyse fonctionnelle technique/produit (ou interne) - Exigences sur les livrables et démarches de mise en oeuvre.
- NF (2000) Management par la valeur - Value management. AFNOR
- NF (2014) Management de la valeur - Vocabulaire - Termes et définitions. AFNOR
- NF EN 13306 (2010) Maintenance - Terminologie de la maintenance. AFNOR

- Nudurupati SS, Bititci US, Kumar V, Chan FTS (2011) State of the Art Literature Review on Performance Measurement. *Comput Ind Eng* 60:279–290. doi: 10.1016/j.cie.2010.11.010
- O2M (2009a) Spécifications de la disposition 3D (liens électriques, connexions, routage 3D...) des composants électroniques dans l'environnement Catia.
- O2M (2008) Méthodologie de prédimensionnement.
- O2M (2009b) Méthodologie : spécification du besoin.
- O2M (2009c) Définition du cahier des charges type d'un produit mécatronique.
- Oakes J, Botta R, Bahill AT (2006) Technical Performance Measures. In: *Proceedings of the 16th Annual International Symposium of the International Council on Systems Engineering (INCOSE)*. Orlando, Florida, USA,
- Oberkampff WL, Barone MF (2006) Measures of agreement between computation and experiment: Validation metrics. *Journal of Computational Physics* 217:5–36. doi: 10.1016/j.jcp.2006.03.037
- Ooshima M, Masukata S (2013) Design methodology for miniaturization of motor equipped in a fluid intermittent control system. In: *2013 International Conference on Electrical Machines and Systems (ICEMS)*. pp 1194–1197
- Pahl G, Beitz W, Feldhusen J, Grote K-H (2007a) *Engineering Design - A Systematic Approach*, Third edition. Springer, London
- Pahl G, Wallace K, Blessing L (2007b) *Engineering design a systematic approach*. Springer, London
- Pailhes J, Colligan A, Sebastian P, Quirante T (2011) *Modélisation et optimisation en conception multi-objectifs*. Besançon, France,
- Park G-J (2007) *Analytic methods for design practice*. Springer, London
- Penas O, Plateaux R, Choley J-Y, Rivière A (2009) The different complexity levels in mechatronic design process. Fez, Morocco,
- Pimmler TU, Eppinger SD (1994) Integration analysis of product decompositions. Alfred P. Sloan School of Management, Massachusetts Institute of Technology Cambridge, MA
- Plateaux R, Penas O, Mhenni F, et al (2009) Introduction of the 3D Geometrical Constraints in Modelica. Como, Italy, pp 526 – 530
- Poels G, Dedene G (1999) DISTANCE: a framework for software measure construction. DTEW Research Report 9937 1–47.

- Pouget J-P (1992) Modélisation géométrique. Institut de recherche sur l'enseignement des mathématiques P, Paris Nord, Université Paris XIII
- Poulin JS, Caruso JM (1993) A reuse metrics and return on investment model. In: , Selected Papers from the Second International Workshop on Software Reusability, 1993. Proceedings Advances in Software Reuse. pp 152–166
- Preparata FP, Shamos MI (1985) Computational Geometry. Springer New York, New York, NY
- Rechtin E, Maier MW (2000) The Art of Systems Architecting. CRC Press
- Reynaerts D, Peirs J, Van Brussel H (1998) A mechatronic approach to microsystem design. IEEE/ASME Transactions on Mechatronics 3:24–33. doi: 10.1109/3516.662865
- Roedler GJ, Lockheed M, Cheryl J (2005) Technical Measurement. INCOSE
- Roumizadeh B, Choley JY, Plateaux R, et al (2009a) An overview on the ECAD/MCAD data exchange: Focus on STEP. University of Strathclyde, Glasgow, UK,
- Roumizadeh B, Choley JY, Plateaux R, et al (2009b) Pre-designing an electronic card using a multi-domain models approach with DYMOLA. Como, Italy, pp 587–592
- Roumizadeh B, Choley J-Y, Rivière A, et al (2009c) A 3D logic modeling for 3D electronic circuits (3D-MID). Marseille,
- Roussel G (1994) Optimisation du placement de formes irrégulières sur matières planes. Application à l'industrie de la confection = Optimization of irregular shapes lay-planning. Application to the garment industry. Université Lille1 - Sciences et Technologies
- Roy B (2005) Paradigms and Challenges. In: Multiple Criteria Decision Analysis: State of the Art Surveys. Springer New York, pp 3–24
- Rudolph S (1995) A methodology for the systematic evaluation of engineering design objects. University of Stuttgart
- Sabre R (2007) Plans d'expériences - Méthode de Taguchi.
- Samin JC, Bröls O, Collard JF, et al (2007) Multiphysics modeling and optimization of mechatronic multibody systems. Multibody Syst Dyn 18:345–373. doi: 10.1007/s11044-007-9076-0
- Sant'Anna C, Figueiredo E, Garcia A, Lucena CJP (2007) On the Modularity of Software Architectures: A Concern-Driven Measurement Framework. In: Oquendo F (ed) Software Architecture. Springer Berlin Heidelberg, pp 207–224
- Scaravetti D (2004) Formulation préalable d'un problème de conception, pour l'aide à la décision en conception préliminaire. Ecole nationale supérieure d'arts et métiers - ENSAM

- Seffah A, Donyaee M, Kline RB, Padda HK (2006) Usability measurement and metrics: A consolidated model. *Software Qual J* 14:159–178. doi: 10.1007/s11219-006-7600-8
- Serré P (2000) Coherence de la spécification d'un objet de l'espace euclidien a n dimensions. Ecole Centrale Paris
- Siddique Z, Rosen DW, Wang N (1998) On the applicability of product variety design concepts to automotive platform commonality. In: ASME Design Engineering Technical Conferences-Design Theory and Methodology.
- Smailagic A, Siewiorek DP, Anderson D, et al (1995) Benchmarking an interdisciplinary concurrent design methodology for electronic/mechanical systems. In: Proceedings of the 32nd annual ACM/IEEE Design Automation Conference. ACM, pp 514–519
- Smith JR (2013) The 4Rs of Metric System Design. <http://www.educause.edu/ero/article/4rs-metric-system-design>.
- Stone RB, Wood KL, Crawford RH (2000) A heuristic method for identifying modules for product architectures. *Design studies* 21:5–31.
- Suh NP (1990) The principles of design. Oxford University Press, New York
- Summers JD, Eckert C, Goel A, others (2013) Function in Engineering: Benchmarking Representations and Models. In: DS 75-2: Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies, Vol. 2: Design Theory and Research Methodology, Seoul, Korea, 19-22.08. 2013.
- Summers JD, Shah J. (2010) Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability. *Journal of Mechanical Design* 132:11. doi: 10.1115/1.4000759
- Summers J, Vargas-Hernandez N, Summers JD, et al (2001) Comparative Study of Representation Structures for Modeling Function and Behavior of Mechanical Devices. In: Behavior of Mechanical Devices", 21st Computers and Information in Engineering Conference, Design Engineering Technical Conference. ASME Press,
- Tahan M, Touil A, Vareille J, Le Parc P (2011) La méthode de développement en X, un autre point de vue sur le cycle de vie. In: 20ème Congrès Français de Mécanique. Besnçon, pp 448–453
- Thouviot S (2013) Optimisation multi-critères d'un système mécatronique en intégrant les problèmes vibro-acoustiques. Ecole Centrale - SUPMECA
- Tinsson W (2010) Plans d'expérience: constructions et analyses statistiques. Springer, Heidelberg ; New York
- Tomizuka M (2002) Mechatronics: from the 20th to 21st century. *Control engineering practice* 10:877–886.

- Tonnellier E, Terrien O (2012) "Rework: Models and Metrics." In: Hammami O, Krob D, Voirin J-L (eds) Complex Systems Design & Management. Springer Berlin Heidelberg, pp 119–131
- Towar A (2008) Quality Function Deployment -QFD). <http://www3.nd.edu/~atovar/ame30362f08/QFD.htm>.
- Toyota Motor Corporation (2003) Toyota Hybrid System - THS II.
- Trabelsi H, Yvars P-A, Louati J, Haddar M (2013) CSP approach and interval computation for the coupling between static and dynamic requirements in the preliminary design of a compression spring. European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique 22:183–197. doi: 10.1080/17797179.2013.820888
- Turki S (2008a) Ingénierie système guidée par les modèles : Application du standard IEEE 15288, de l'architecture MDA et du langage SysML à la conception des systèmes mécatroniques. Laboratoire LISMMMA - Supmeca
- Turki S (2008b) Model-Driven Engineering: Application of the IEEE 15288 standard, the MDA architecture and the SysML language to mechatronics systems design. Université du Sud Toulon Var
- Ullman (2000) the ideal engineering decision support system .PDF - theidealenginsyste1.pdf. AutoDesk
- Ulrich K, Eppinger S (2011) Product Design and Development, Édition : 5. McGraw-Hill Higher Education, New York
- van der Meiden HA, Bronsvoort WF (2007) Solving topological constraints for declarative families of objects. Computer-Aided Design 39:652–662. doi: 10.1016/j.cad.2007.05.013
- VDI-Fachbereich Produktentwicklung und Mechatronik (2004) Design methodology for mechatronic systems. VDI-Gesellschaft Produkt- und Prozessgestaltung
- Vergeest JSM, Spanjaard S, Song Y (2003) Directed mean Hausdorff distance of parameterized freeform shapes in 3D: a case study. Visual Comp 19:480–492. doi: 10.1007/s00371-003-0213-3
- Verries J (2010) Approche pour la conception de systèmes aéronautiques innovants en vue d'optimiser l'architecture. Application au système portes passager. Université Paul Sabatier - Toulouse III
- Wani MF, Gandhi OP (1999) Development of maintainability index for mechanical systems. Reliability Engineering & System Safety 65:259–270.

- Warniez A, Penas O, Plateaux R, Soriano T (2014a) SysML Geometrical Profile for Integration of Mechatronic Systems. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Besançon, France,
- Warniez A, Penas O, Plateaux R, Soriano T (2014b) SysML geometrical profile for integration of mechatronic systems. In: 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). pp 709–714
- Warniez A, Penas O, Soriano T (2012) About metrics for integrated mechatronic system design. Saint Ouen, France, pp 450 – 457
- WRAP (2008) Recyclability efficiency metric - Development of options for setting client procurement requirements for recyclability in construction projects.
- Xijuan L, Yinglin W, Shouwei J (2003) A metrics based task analysis model for design review planning. *Design Studies* 24:375–390. doi: 10.1016/S0142-694X(02)00039-X
- Yang M (2008) Hausdorff dimension of the image of additive processes. *Stochastic Processes and their Applications* 118:681–702. doi: 10.1016/j.spa.2007.05.006
- Yassa S, Sublime J (2013) A genetic algorithm for multi-objective optimisation in workflow scheduling with hard constraints. *International Journal of MetaHeuristics*. doi: 10.1504/IJMHEUR.2013.058475
- Zerelli M (2014) Systèmes mécatroniques à paramètres variables : analyse du comportement et approche du tolérancement. Ecole centrale de Paris
- Zhang Y, Gershenson JK (2003) An initial study of direct relationships between life-cycle modularity and life-cycle cost. *Concurrent Engineering* 11:121–128.
- Zwingmann X (2005) Modèle d'évaluation de la fiabilité et de la maintenabilité au stade de la conception. Université Louis-Pasteur

ANNEXES

ANNEXE 1 : RECHERCHE DE CONCEPT

Comme indiqué sur la Figure 1-10 du Chapitre 1, l'analyse du besoin et l'analyse fonctionnelle permettant la formalisation des exigences constituent les points d'entrée pour le premier niveau d'architecture : l'architecture fonctionnelle. Delà la conception architecturale peut débuter

Ces phases préalables font souvent appel à différentes méthodes, suivant la culture d'entreprise ou des concepteurs et le niveau de raffinement requis, et correspondent à ce que les concepteurs appellent communément la phase d'analyse externe. Nous rappelons ici rapidement les méthodes classiquement utilisées pour cette phase préalable d'émergence de concepts et d'analyse fonctionnelle pour la constitution d'un cahier des charges fonctionnel. Les méthodes peuvent différer suivant l'objectif de la conception.

Brainstorming

Cette technique de résolution créative, conçue en 1940 par Alex Osborn, est généralement utilisée en petit comité, de façon plutôt informelle pour faire émerger les concepts de produits à développer. Cela peut démarrer d'un croquis, de quelques mots clefs, d'une idée qui peut apparaître initialement comme farfelue... Le principe est de mettre en présence des personnes, de services, compétences, domaines variés pour favoriser au maximum la créativité. A cette étape, la faisabilité finale des idées n'est pas un objectif, les personnes donnent juste leurs idées personnelles. Cette approche permet d'impliquer toutes ces personnes dans le système à développer en favorisant la cohésion de groupe et en stimulant la motivation de chacun à mettre tout en œuvre pour la réussite du produit. Il s'avère pourtant parfois délicat de rendre cette méthode vraiment productive, sans un certain accompagnement, guidage et autres règles communes aux processus collaboratifs. Un animateur « externe » peut alors s'avérer nécessaire.

APTE

Les concepteurs français utilisent souvent la méthode spécifique d'analyse fonctionnelle et d'analyse de la valeur APTE® (APplication des Techniques d' Entreprise). Cette pratique a été standardisée par la norme NF X50-100 (NF 2011) et son efficacité a été validée lors du programme européen de lancement d'Ariane. APTE constitue la première étape de l'analyse fonctionnelle séquentielle : l'analyse externe pour établir les exigences « client ». Une fois que les frontières du système sont définies, en séparant le système de son environnement, l'outil le plus puissant de la méthode APTE® est le graphe des interacteurs. Il fournit au concepteur toute l'aide nécessaire pour exprimer les besoins, au travers d'une liste d'exigences, quantifiées par des critères et des flexibilités. Un tel graphe doit être réalisé pour chaque phase du cycle en V du système : conception, fonctionnement normal, maintenance, recyclage...).

APTE constitue la première phase de conception débouchant sur l'édition du cahier des charges fonctionnel.

Le diagramme des interacteurs également appelé « pieuvre » permet de recenser les fonctions de service d'un produit grâce à un outil graphique. Ce diagramme est constitué du produit, au centre, et, autour, des éléments de son environnement (milieu extérieur). Il fait apparaître les relations (fonctions) entre le produit et le milieu extérieur (sans oublier les contraintes de haut niveau comme le coût, le développement durable....

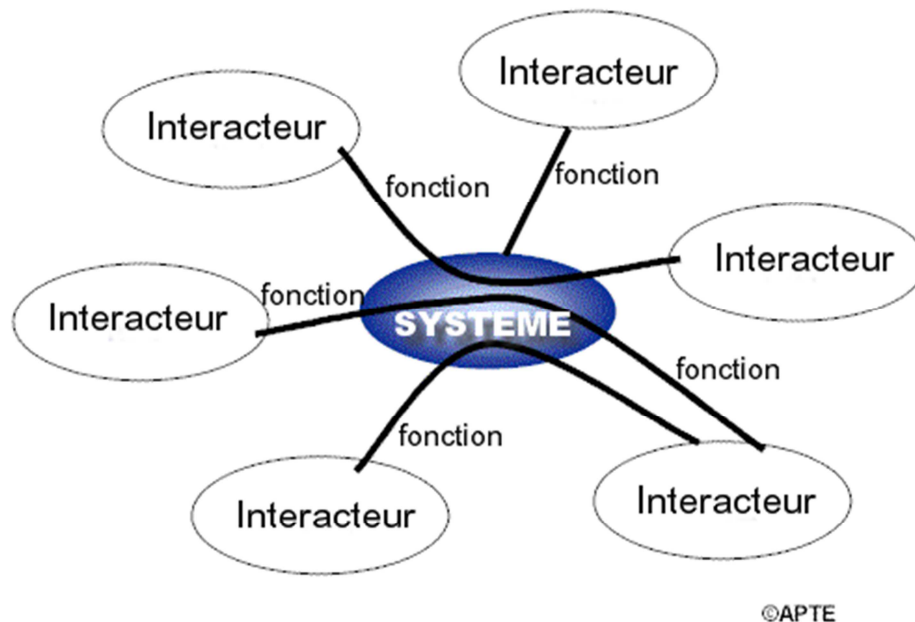


Figure A1-0 : Exemple théorique d'un diagramme "pieuvre"

Ces relations correspondent au service rendu par le produit et permettent d'élaborer un cahier des charges.

Deux types de fonctions sont en jeu :

- les fonctions principales (également appelées Fonctions de Service d'Usage) : ce sont les fonctions pour lesquelles le produit est élaboré, donc en fait celles qui pourraient répondre aux exigences de l'utilisateur. On les dessine par des liens entre deux éléments de l'environnement, liens passant par le système ;
- les fonctions contraintes (également appelées Fonctions de Service d'Adaptation) : elles sont un lien entre le produit et un élément de l'environnement. Elles naissent d'une contrainte imposée par un élément extérieur, de l'existence d'un produit déjà existant ou encore d'une exigence particulière de l'utilisateur voire de la présence de normes et de législations.

L'expression des fonctions peut faire apparaître une notion quantifiable, qui par développement donnera des critères chiffrés du cahier des charges fonctionnel. Ils seront exprimés dans une table de recueil des exigences où pour ces fonctions, chaque critère sera alors affecté d'un niveau de flexibilité (souvent un chiffre entre 0 et 3) qui indique l'écart admis par rapport à la spécification considérée (0 : non négociable).

Nous citerons également les méthodes basées sur l'approche MBSE et souvent implémentées en langage SysML, dont une a été détaillée dans le 0, §1.2.2.3.

ANNEXE 2 : RECHERCHE ET EMERGENCE D'ARCHITECTURE CANDIDATES

La Méthode TRIZ

TRIZ est l'acronyme russe de la théorie de résolution des problèmes inventifs. C'est une méthode de résolution des problèmes techniques inventée par Altshuller en 1946 (Altshuller 1984). Durant cette analyse, il a focalisé son attention sur les processus d'obtention de solutions, et a ainsi constitué les prémices d'une base de connaissances. La méthode TRIZ considère donc que les problèmes rencontrés dans la conception d'un nouveau produit présentent des analogies avec d'autres. Seulement 1% des solutions sont réellement innovantes. Elle favorise la créativité ou la simulation de concepts innovants tout en guidant le concepteur à partir de « fils rouges ».

La méthode de résolution des problèmes par la démarche TRIZ (Figure A2-1) se décompose en étapes suivantes :

- Description de la situation initiale ;
- Identification du problème à résoudre ;
- Formulation d'une solution idéale ;
- Formulation d'une « solution physique » : sous sa forme générale permettant de se rapprocher de la solution idéale ;
- Formulation d'une « solution technique » : principe de réalisation de la solution physique ;
- Formulation d'une solution spécifiée, description complète de la solution, calcul des caractéristiques de la solution.

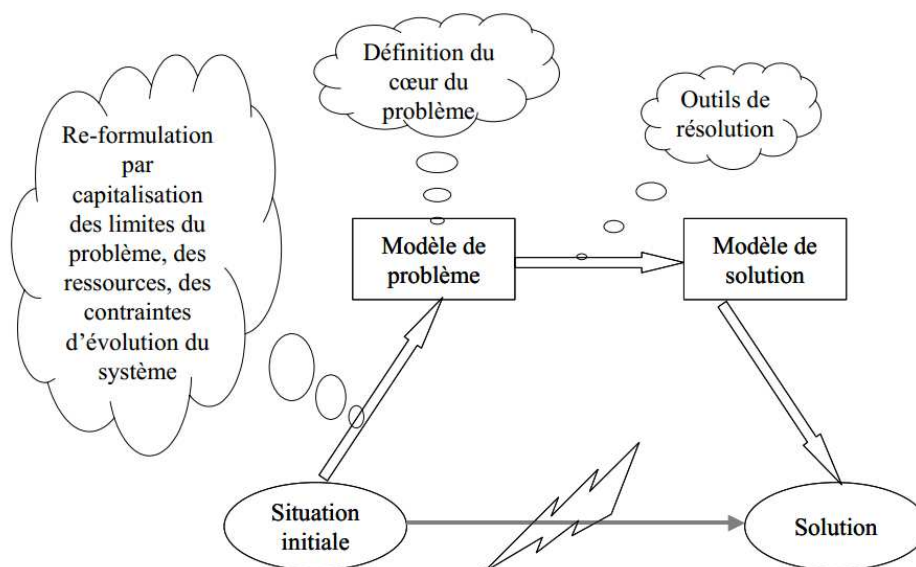


Figure A2-1 : Démarche générale de la méthode TRIZ (Dubois 2004)

Concrètement, TRIZ permet de résoudre les contradictions apparaissant durant une nouvelle conception. Les outils de TRIZ sont particulièrement utilisés en France dans l'industrie

automobile et dans l'aéronautique. Les outils de cette méthode sont principalement basés sur un algorithme appelé ARIZ. Il permet de résoudre des mini-problèmes fortement contraints.

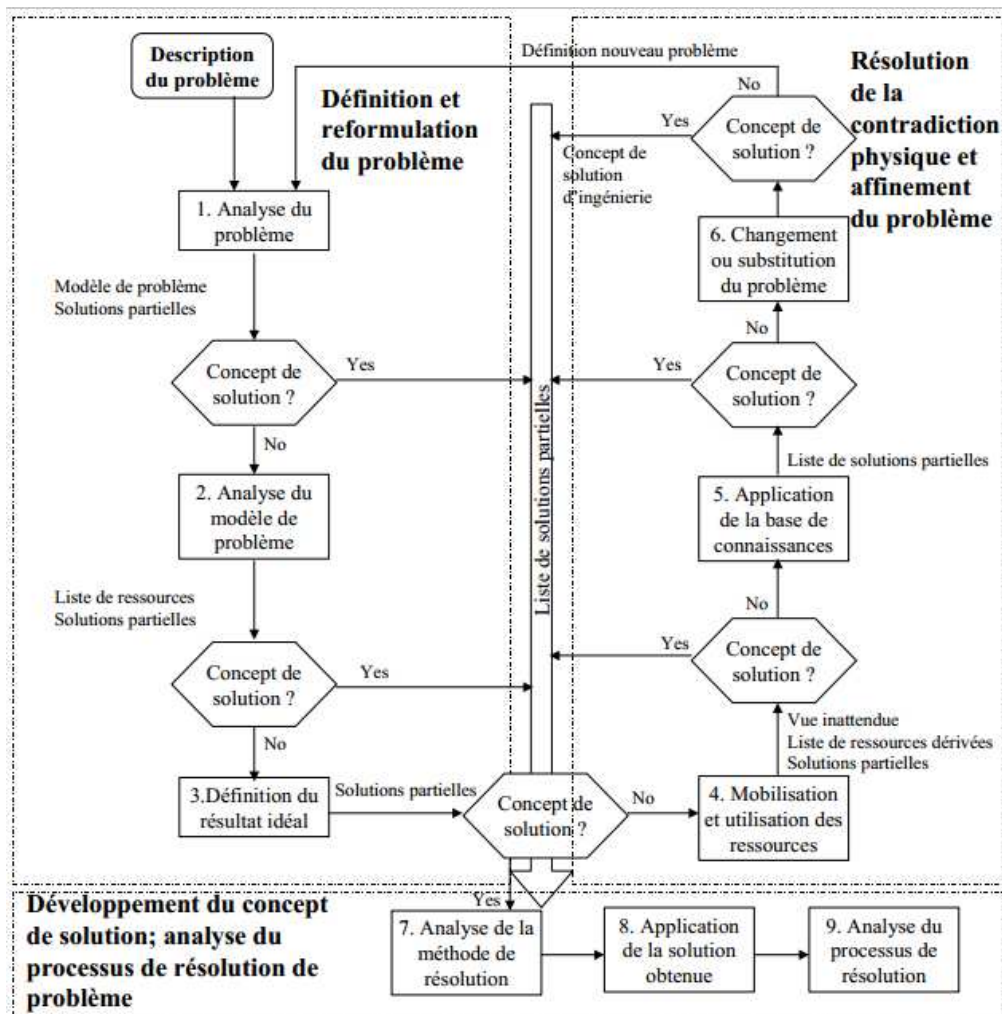


Figure A2-2 : Algorithme d'ARIZ (Dubois 2004)

L'apport de la méthode TRIZ est le fait de centrer le processus de conception sur la résolution de problème (Dubois 2004) et donne des directions pour la recherche des solutions basées sur des modèles d'ingénierie (Figure A2-2).

Concernant la conception mécatronique, Casner et al. (Casner et al. 2013) font appel au Raisonnement à Partir de Cas (CBR : Case Based Reasoning), inspiré de la méthode TRIZ. Leur approche est basée sur l'intelligence artificielle. Un système CBR repose sur une base de cas où un cas possède une description et une solution. Pour résoudre un problème, le système recherche des cas similaires au problème posé puis fournit une solution adaptée qui doit être validée. Pour finir, il rajoute le nouveau cas (avec sa description et sa solution) à la base de données des cas. Cette méthode peut être utilisée dans la recherche de solution d'architecture si l'on considère la conception d'un système mécatronique comme un problème à résoudre par des algorithmes.

ANNEXE 3 : FORMALISATION / MODELISATION DES ARCHITECTURES

SADT/IDEF⁵¹

La modélisation SADT (Structured Analysis and Design Technique) est une méthode d'analyse et de conception fonctionnelle descendante par niveaux successifs décrivant le système, qui a été introduite en 1969 par la société Softech Inc. En 1974, l'US Air Force a alors développé les méthodes/langages IDEF (ICAM DEFinition) issus de son projet ICAM, pour améliorer la productivité en fabrication, qui sont devenues maintenant des standards (FIPS 1993a; FIPS 1993b; IEEE 1998).

Ces méthodes référencées par numéro, et dont SADT, est la base sont utilisées de la manière suivante^{52,53}:

- IDEF-0 : architecture fonctionnelle (et logique), correspond aux actigrammes SADT
- IDEF-1 : architecture organique et de données, correspond aux datagrammes SADT
- IDEF-2 : architecture intégrant le temps (aspects dynamiques)
- ... IDEF14 : la famille de langages de modélisation IDEF couvre désormais une large gamme d'utilisation, de la modélisation fonctionnelle aux données, simulation, analyse et conception orientées objet et acquisition des connaissances.

Ainsi, les grands principes pour décrire la méthode SADT, utilisée pour les architectures fonctionnelles et organiques, sont :

- initialement, le système est vu comme une boîte noire, avec une introduction graduelle des détails : une boîte est décomposée (ouverture de la boîte) en sous-boîtes moins complexes : on passe successivement du plus abstrait/général au plus particulier.
- Suivant que la modélisation concerne l'architecture fonctionnelle (actigrammes) ou organique (datagrammes), on retrouve une dualité (voir Figure A3-1) :
 - entre les actigrammes, où les boîtes sont des activités (verbes) et les flèches les données (noms) ;
 - et les datagrammes, où les boîtes sont des données et les flèches les activités.
- La position des flèches sur une boîte est significative (Figure A3-2) :
 - Pour les actigrammes (en ce qui concerne l'architecture fonctionnelle, les activités correspondent aux fonctions et les données aux flux) :
 - Entrées : données transformées par l'activité en sortie
 - Sorties : données créées par l'activité
 - Contrôles : données dont la présence contraint l'activité ou les données non modifiées par l'activité
 - Mécanismes : support qui effectue l'activité (« au moyen de ... »)
 - Datagrammes (en ce qui concerne l'architecture organique, les données correspondent aux organes, les activités aux fonctions)
 - Entrées : activités génératrices de données

⁵¹ <http://adrcomptefreeflo.free.fr/Nouveau%20dossier/IDEF-SADT.pdf>

⁵² <http://www.idef.com/>

⁵³ <http://en.wikipedia.org/wiki/IDEF>

- Sorties : activités utilisatrices de données
- Contrôles : activités dont le résultat influe la création ou l'utilisation des données
- Mécanismes : unités de stockage des données

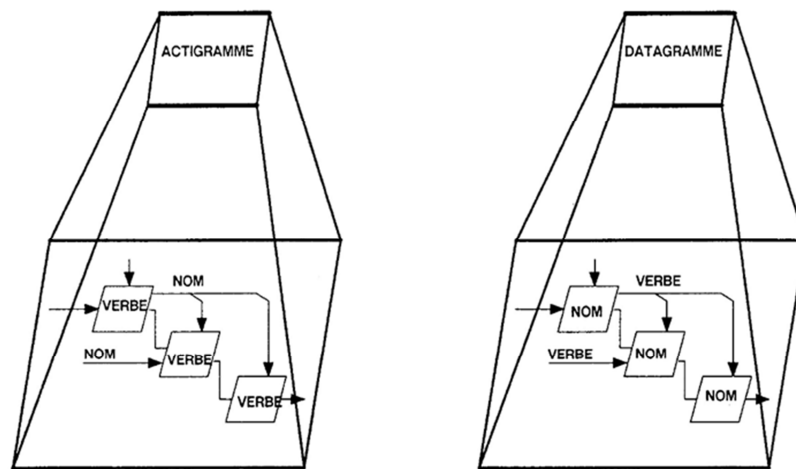


Figure A3-1 : Architecture fonctionnelle (actigramme) et organique (datagramme) avec SADT/IDEF⁵¹

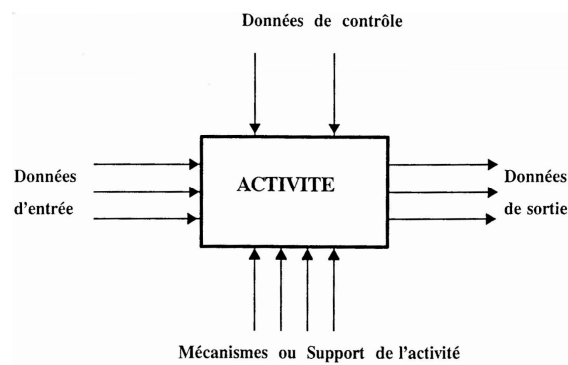
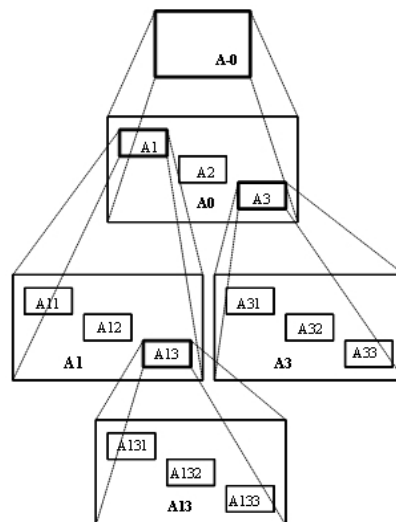


Figure A3-2 : Description des flèches d'un diagramme⁵¹

Ainsi, une fonction est représentée par une « boîte » SADT (actigramme) qui modélise une fonctionnalité et des flèches qui représentent les flux d'échanges entre le système et son environnement ou pour situer une « boîte » par rapport à d'autres « boîtes ». Ces flèches sont des flux de matières, des flux d'énergies, des flux d'informations et des flux de sorties. D'autre part, une « boîte » est représentée par un rectangle contenant un verbe et un label d'identification A_{ij} (Actigramme n°j du niveau i ; Figure A3-3). A-0 correspond à l'actigramme de la fonction globale, non décomposée. A1, A2 et A3 sont en fait A01, A02, A03 (0 est omis par définition).

Figure A3-3: Exemple d'architecture hiérarchique fonctionnelle⁵¹

Les avantages sont la production d'une structure hiérarchisée par niveau, permettant une décomposition analytique de la complexité du système, et l'intemporalité de ses diagrammes.

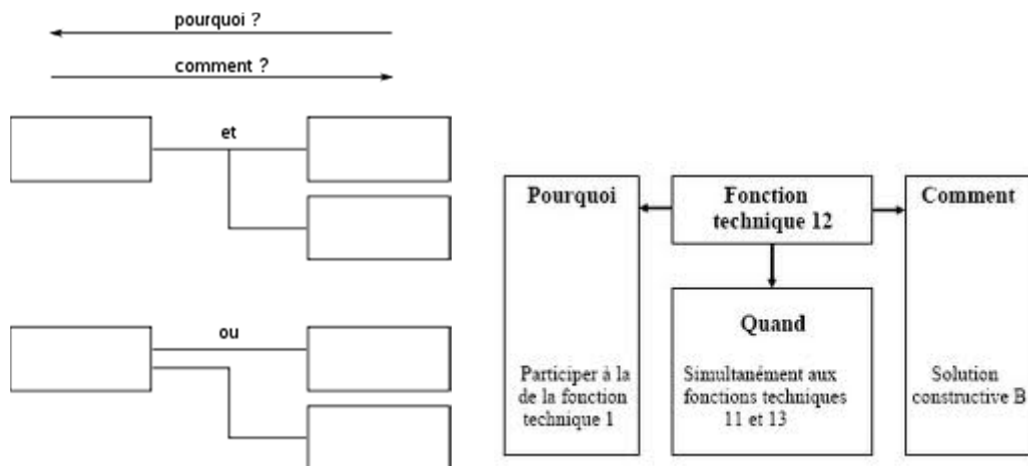
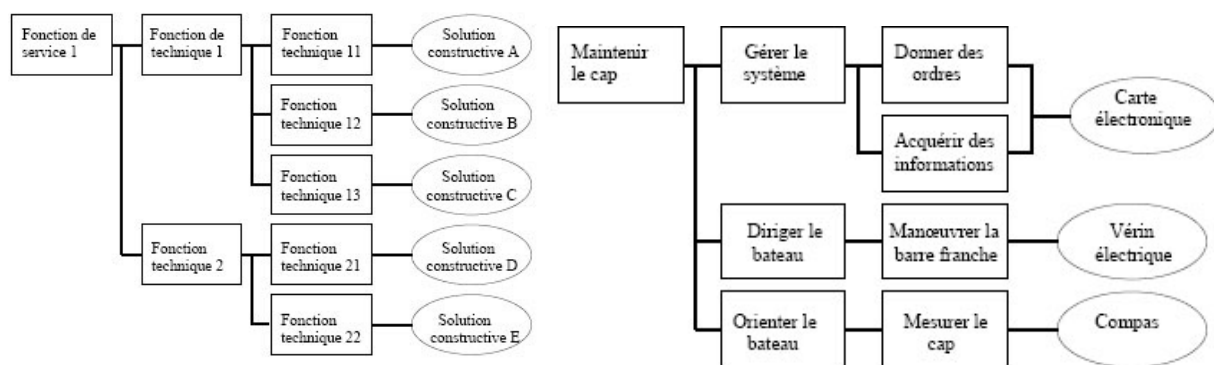
Les inconvénients de cette modélisation sont qu'elle n'adresse pas tous les niveaux d'architecture, qu'elle n'intègre pas de représentation séquentielle, ni d'opérateur booléen, enfin la vue globale n'est accessible qu'au plus haut niveau.

FAST⁵⁴

La technique FAST (Functionnal Analysis System Technique), développée en 1964 par C. Bytheway, maintenant décrite dans la norme NF EN 12973 (NF 2000), permet de représenter, de gauche à droite, les fonctions de services traduites en fonctions techniques puis en solutions techniques. Elle peut donc adresser tous les niveaux d'architecture, de fonctionnelle à physique, même si généralement les solutions techniques sont plutôt des organes que des composants strictement définis (Figure A3-4).

Le concepteur développe les fonctions de service en répondant à la question « pourquoi ? », puis il répond à la question « comment ? » en concevant les fonctions techniques, et enfin il matérialise le système avec des solutions techniques. Le processus de recherche de solutions technologiques s'appuie généralement sur la méthode du brainstorming (voir Annexe 1).

⁵⁴ http://www.value-eng.org/pdf_docs/monographs/FAbasics.pdf

Figure A3-4: Principe des architectures FAST⁵⁵Figure A3-5 : Exemple d'une description FAST⁵⁶

- Les **fonctions de service** : constituent une relation entre le système et le milieu extérieur, elles traduisent l'action attendue ou réalisée par le produit pour répondre à un élément du besoin d'un utilisateur donné. Il faut souvent plusieurs fonctions de service pour répondre à un besoin. Dans une étude donnée, leur énumération et leur formulation qualitative et quantitative résultent de l'analyse du besoin à satisfaire et le décrivent d'une manière nécessaire et suffisante. Il existe deux types de fonctions de service :
 - les fonctions principales, correspondant au service rendu par le système pour répondre aux besoins.
 - les fonctions contraintes, traduisant des réactions, des résistances ou des adaptations à des éléments du milieu extérieur.
- Les **fonctions techniques** : sont internes au produit, elles sont choisies par le constructeur dans le cadre d'une solution, pour assurer une fonction de service.

Les avantages de cette technique sont qu'elle permet de modéliser des architectures avec des fonctions logiques comme le « et » et le « ou » ; qu'elle offre une conception créative en proposant des alternatives de réponse pour chaque fonction. Le modèle FAST constitue un ensemble de données permettant une bonne connaissance du système. Enfin, cet outil est normalisé en France (norme NF EN 1325-1(NF 2014)) et permet donc une diffusion assez large.

⁵⁵ <http://intratek.legtux.org/Intra-Tek/Pages/3/4-Projet3-Maison/Pages/3-Fast/index.htm>

⁵⁶ http://fr.wikipedia.org/wiki/Function_Analysis_System_Technique

En ce qui concerne les inconvénients, cette modélisation ne permet pas d'avoir des informations sur les flux échangés entre les différentes fonctions et solutions et ne permet pas non plus de « sélectionner/marker » les « mauvaises » alternatives.

Langages de modélisation pour l'architecture logique

Le niveau logique « correspond à la définition des entités logiques qui représentent les exigences du système : la performance, l'évolution dans le temps, les flux d'énergies, les données échangées, etc. » (Hammadi 2012a). Les entités logiques sont organisées dans une architecture logique avant que les éléments physiques soient définis. Cette architecture modélise le comportement du système à l'aide d'un ensemble d'équations et permet donc la simulation. Les outils de modélisation pour l'architecture logique permettent également de simuler ces différentes architectures candidates pour vérifier lesquelles présentent les performances attendues. Ainsi, les résultats de simulation permettent d'avoir une cartographie des architectures qui permettent de répondre aux exigences, avec en plus la ou les valeurs admissibles des paramètres de conception pour que l'architecture évaluée reste valide.

Les principaux langages et logiciels permettant de modéliser et simuler l'architecture logique sont les suivants :

- *VHDL*⁵⁷ : langage standard de modélisation (norme IEEE 1076.1-1999)(IEEE 1999b) qui dérive du langage de description matériel VHDL (norme IEEE 1076-1993) (IEEE 2000). Il est principalement utilisé pour modéliser et simuler des systèmes électriques et électroniques, les circuits intégrés complexes et des systèmes à signaux mixtes.
- *Modelica*⁵⁸ : langage orienté objet qui a été développé pour la modélisation et la simulation des systèmes physiques. Il intègre la description mathématique déclarative des modèles. Certains outils l'implémentant proposent également l'approche graphique de modélisation à l'aide des diagrammes blocs (Hammadi 2012a), et de nombreuses bibliothèques thématiques ont été développées. Il est principalement utilisé pour la modélisation des systèmes mécaniques et mécatroniques.
- *Simulink*⁵⁹ : logiciel de modélisation et de simulation multi-domaine des systèmes. Il propose un éditeur graphique, un ensemble personnalisable de bibliothèques de blocs et des solveurs pour la modélisation et la simulation de systèmes dynamiques. Simulink est intégré à MATLAB®, ce qui permet d'incorporer des algorithmes MATLAB dans les modèles et d'exporter le résultat des simulations vers MATLAB pour compléter les analyses.

Il existe d'autres logiciels de modélisation et simulation des systèmes complexes pour des domaines spécifiques (vibration, thermique, électromagnétisme,...) comme Comsol, Ansys, Flux2D/3D, Inca3D etc.

⁵⁷ <http://en.wikipedia.org/wiki/VHDL>

⁵⁸ <https://www.modelica.org/>

⁵⁹ <http://www.mathworks.fr/products/simulink/>

Langages, méthodes et outils pour les architectures physiques et spatiales

En ingénierie système, le niveau physique est le niveau où les composants sont choisis et caractérisés. Les architectures physiques correspondantes peuvent être modélisées avec les mêmes outils que pour l'architecture organique. Pour l'architecture physique, le concepteur peut également s'appuyer sur la modélisation hiérarchique présentée en *Annexe 4*.

ANNEXE 4 : MODELISATION HIERARCHIQUE

Aujourd'hui, les produits mécatroniques sont constitués de nombreux systèmes, sous-systèmes, modules et composants. Ils se composent d'une grande variété de sous-systèmes de disciplines différentes, avec de nombreuses interconnexions et relations complexes. Afin de maîtriser cette complexité, l'utilisation de modèles avec des « granularités » adaptées à des niveaux hiérarchiques d'abstraction différents semble une approche prometteuse. En effet, concevoir un système, c'est le modéliser à différents niveaux d'abstraction.

Une structure hiérarchique du système s'obtient par la décomposition du système en éléments (composants, sous-systèmes). Pour les systèmes complexes comme les systèmes mécatroniques, elle est utilisée pour définir différentes vues d'un système comme la géométrie, l'agencement spatial, le personnel, les flux de matière, d'énergies, etc. Pour les systèmes mécatroniques, il est notamment important que cette décomposition en sous-systèmes séparés, définisse les frontières de ceux-ci de telle sorte que les interfaces soient évidentes pour tous les ingénieurs des disciplines impliquées. Les modèles du niveau système illustrent alors la dépendance entre les sous-systèmes (qui sont eux-mêmes constitués de solutions de disciplines d'ingénierie différentes). Cette approche fournit une vue multi-niveau de l'ensemble du système à étudier (Hehenberger 2012). Elle est applicable à l'analyse de presque tous les problèmes d'ingénierie et notamment celle des problèmes multi-physiques. En effet, au début de chaque analyse d'un problème d'ingénierie, il est recommandé d'utiliser les modèles les plus simples, avant de parvenir, grâce à la méthode de modélisation hiérarchique, à un modèle exhaustif avec le niveau de raffinement souhaité. La modélisation hiérarchique permet ainsi de modéliser des problèmes complexes et d'effectuer suivant le besoin, un transfert facile des modèles du niveau conceptuel à un niveau de conception plus détaillée.

Dans son article (Kellner and Hehenberger 2014), Kellner explique le fort intérêt de cette modélisation hiérarchique pour les systèmes mécatroniques. Elle permet de partitionner le système étudié en éléments plus simples et plus rapides à traiter, et ainsi de diminuer leur complexité. Les modèles de ces éléments sont ainsi créés plus facilement et peuvent être manipulés indépendamment. Ils peuvent alors être regroupés par niveaux et chaque niveau est lié dans une relation hiérarchique avec un niveau « haut » et un niveau « bas » et possède ses propres entités/modèles. La combinaison de ces relations hiérarchiques dans les modélisations ou encore dans les aspects organisationnels facilite et améliore la conception des systèmes mécatroniques.

Hehenberger (Hehenberger 2012) a montré qu'un système mécatronique peut être divisé en quatre niveaux hiérarchiques, à savoir : le système mécatronique global, décomposé en plusieurs systèmes mécatroniques qui sont constitués de modules mécatroniques, qui eux-mêmes sont constitués de composants spécifiques à un seul domaine ou discipline. Ainsi, dans cette représentation, les niveaux hiérarchiques les plus bas reflètent l'architecture des modules mécatroniques en composants mono-domaines (Figure A4-1).

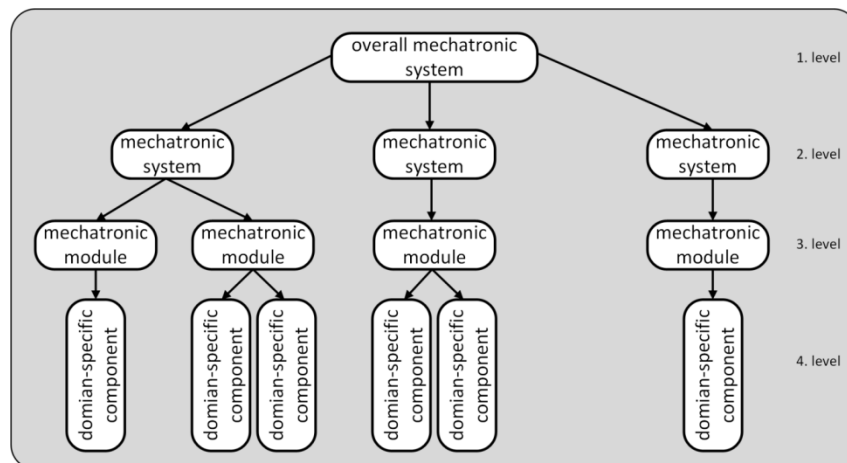


Figure A4-1 : Structure hiérarchique d'un système mécatronique (Kellner and Hehenberger 2014)

Ainsi l'architecte système peut obtenir un aperçu simple d'un système complexe, sa structure sous-jacente, ses modules et ses sous-composants.

Concernant la simulation, chaque modèle, quel qu'en soit le niveau hiérarchique, est composé d'un ensemble de paramètres qui sont, soit des paramètres de conception internes (nécessaires uniquement pour ce modèle dans le niveau concerné), soit des paramètres de conception externes (nécessaires pour les liens avec les autres modèles sur les différents niveaux).

A titre d'illustration, dans (Casner et al. 2013), l'auteur utilise une démarche d'optimisation hiérarchisée pour les systèmes mécatroniques. Il définit 5 étapes : la génération de la structure mécanique, la sélection des actionneurs, le choix des capteurs, la définition de la stratégie de commande, puis la dernière étape est celle de la prise de décision et l'évaluation de la solution. Il prend l'exemple de conception d'une pince de robot (Figure A4-2).

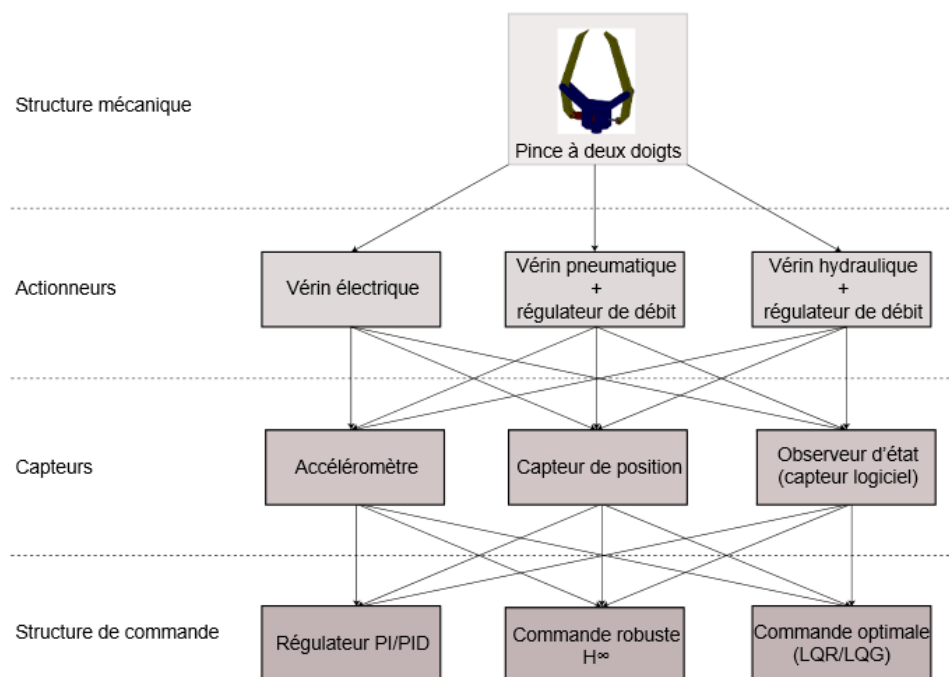


Figure A4-2 : Solutions candidates par l'approche hiérarchique de conception des systèmes mécatroniques selon (Casner et al. 2013)

Cependant, lorsque les composants (mécanique, électronique etc.) et donc leur géométrie sont connus, il devient intéressant d'utiliser des logiciels de CAO (Conception Assistée par Ordinateur) pour agencer ces différents composants dans l'espace, afin d'évaluer les architectures spatiales candidates du système (Chapitre 3). En mécanique, par exemple, l'architecture physique du système technique est matérialisée par des modèles géométriques 2D ou 3D des composants du système qui réalisent la maquette 3D. Les logiciels de CAO utilisés sont par exemple CATIA⁶⁰ et SolidWorks⁶¹ de Dassault Systèmes, FreeCad⁶² (outil libre), TopSolid⁶³ de Missler Software, Creo⁶⁴ de PTC, Nx⁶⁵ de Siemens, etc. En électronique, il existe également des logiciels de CAO permettant de placer les composants sur différents supports (PCB, Flex,...) : Eagle⁶⁶, Orcad⁶⁷, Cadence⁶⁸ etc. Il existe aussi quelques logiciels de CAO pour des domaines plus spécifiques comme l'électrotechnique (Ecodial), l'électromagnétisme (Inca3D).

Enfin, notons que ces logiciels de CAO peuvent également être déjà utilisés au niveau logique avec les volumes d'encombrement attribués pour les organes, afin de formaliser en phase amont les premières contraintes géométriques/volumique du cahier des charges. Avec Catia Systems, un lien dynamique entre la CAO et le modèle logique peut être créé permettant ainsi la mise à jour automatique des propriétés inertielles.

60 <http://www.3ds.com/fr/produits-et-services/catia/>

61 <http://www.solidworks.fr/>

62 <http://www.freecadweb.org/>

63 <http://www.topsolid.fr/>

64 <http://fr.ptc.com/product/creo>

65 http://www.plm.automation.siemens.com/fr_fr/products/nx/

66 <http://www.cadsoftusa.com/>

67 <http://www.orcad.com/>

68 <http://www.cadence.com/us/Pages/default.aspx>

ANNEXE 5 : DEFINITION MATHEMATIQUE

Espace métrique

Un ensemble E muni d'une distance d s'appelle un *espace métrique*.

Ouvert, fermé

Les ouverts

Soit E un espace métrique de distance d . On dit qu'un sous-ensemble U de E est ouvert si pour tout point $x \in U$ il existe une boule ouverte de centre x contenue dans U . D'après un principe général de logique, l'ensemble vide, qui n'a pas d'élément, est donc ouvert. La famille des ouverts d'un espace métrique vérifie les propriétés suivantes :

E et \emptyset sont des ensembles ouverts ;

Toute réunion (finie ou pas) d'ensembles ouverts est un ensemble ouvert ;

Toute intersection finie d'ouverts est un ouvert.

Les fermés

Un sous-ensemble F de E est dit fermé si son complémentaire dans E est un ensemble ouvert. Par exemple toute boule fermée $B_f(x_0, r)$ est un ensemble fermé ; en effet, si $x \notin B_f(x_0, r)$, on a :

$$B(x, r') \cap B_f(x_0, r) = \emptyset$$

Pour $r' = d(x_0, x) - r > 0$, comme cela résulte facilement de l'inégalité triangulaire.

De même la famille des fermés d'un espace métrique vérifie les propriétés suivantes :

E et \emptyset sont des ensembles fermés ;

Toute intersection (finie ou pas) d'ensembles fermés est un ensemble fermé ;

Toute réunion finie de fermés est un fermé.

Compact

Soit K une partie d'un espace métrique E . K est compact s'il vérifie la propriété de Borel-Lebesgue : si tout recouvrement de K par des ouverts on peut extraire un sous recouvrement fini.

Boule unité ouvert, Boule unité fermé

Boule ouverte

Dans un espace métrique E de distance d , on appelle boule ouverte de centre $x_0 \in E$ et de rayon $r > 0$, l'ensemble des points de E dont la distance à x_0 est strictement inférieure à r , soit :

$$B(x_0, r) = \{x \in E \mid d(x_0, x) < r\}$$

La boule unité ouverte est la boule ouverte $B(0,1)$ centrée à l'origine et de rayon 1.

Boule fermée

De la même manière, on définit la boule fermée de centre x_0 et de rayon r par :

$$B_f(x_0, r) = \{x \in E \mid d(x_0, x) \leq r\}$$

La boule unité fermée est la boule fermée $B_f(0,1)$ centrée à l'origine et de rayon 1.

ANNEXE 6 : CONSTRUCTION D'UNE ENVELOPPE CONVEXE 3D

La construction de l'enveloppe convexe est tirée d'un sujet proposé par Philippe Chassignet⁶⁹

Etant donné un ensemble de points dans \mathbb{R}^3 , leur enveloppe convexe est la frontière du plus petit convexe contenant ces points.

L'algorithme est le suivant :

1. Recherchons une première face :
 - Nous prenons comme premier point p_1 , celui dont la coordonnée z est minimale.
 - Nous prenons comme second point p_2 , celui qui minimise l'angle entre la droite p_1p_2 et le plan d'équation ($z=cste$) passant par p_1 .
 - Pour le troisième point, nous prenons l'arête $\overline{p_1p_2}$ que nous venons de construire, puis nous définissons :
 - n , un vecteur unitaire orthogonal à l'arête $\overline{p_1p_2}$,
 - k , le vecteur unitaire orthogonal à n et à l'arête $\overline{p_1p_2}$
 - p , un point sur l'arête $\overline{p_1p_2}$.

Le troisième point est un point q de l'ensemble initial, n'appartenant pas à l'arête $\overline{p_1p_2}$ et optimisant le critère suivant :

$$\frac{\overline{pq} \cdot k}{\sqrt{(\overline{pq} \cdot n)^2 + (\overline{pq} \cdot k)^2}} \quad (2.1)$$

Ce critère est le cosinus de l'angle d'ouverture entre l'arête $\overline{p_1p_2}$ et le plan qui s'appuie sur cette arête et passe par q . La face ainsi créée sera celle déterminée par les deux points extrémités de $\overline{p_1p_2}$ et par le point q qui minimise ce critère.

Une fois la première face créée, nous obtenons alors 3 arêtes qui servent à initier un ensemble de « bords ». Un bord est une arêtes, qui provisoirement, n'est encore liée qu'à une seule face.

2. Puis nous calculons la deuxième face sur l'une des arêtes a de la face f_1 , pour cela :
 - le vecteur unitaire n , est orthogonal à la face f_1 , et cette fois-ci orienté vers l'intérieur du convexe,
 - le vecteur unitaire k est orthogonal à n et à l'arête a et orienté vers l'intérieur de la face f_1 ,
 - p , un point sur l'arête a .

⁶⁹<http://www.enseignement.polytechnique.fr/informatique/profs/Georges.Gonthier/pi98/env3D.html#Preparata-Shamos>

Pour créer la deuxième face, nous prenons un point q qui optimise le critère (2.1). La nouvelle face f_2 recherchée est celle déterminée par les deux points extrémités de a et par le point q qui minimise ce critère.

Si plusieurs points minimisent ce critère, on devra écarter ceux qui se trouvent à l'intérieur des triangles construits avec d'autres points.

3. Tant qu'il reste (au moins) une arête de bord, nous refaisons le point 2.
4. Pour obtenir le polyèdre convexe minimum en nombre de faces, il faut fusionner toutes les paires de faces qui sont coplanaires et qui partagent une arête.

ANNEXE 7 : PROGRAMME DE CONSTRUCTION D'UNE ENVELOPPE CONVEXE 3D EN LANGAGE JAVA

Cet algorithme est tiré de la source suivante :

[HTTP://WWW.EECS.TUFTS.EDU/~MHORN01/COMP163/ALGORITHM.HTML](http://www.eecs.tufts.edu/~mhorn01/comp163/algorithm.html)

```
/*
 * @(#)Viewer.java
 *
 * Final Project
 * COMP163: Computational Geometry
 * Tufts University, Department of Computer Science
 * Spring 2004
 */
import j3d.*;
import hull.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;

/**
 * Main applet that controls visualization and user interaction.
 * This part of the code is essentially throw-away, so browse at
your
 * own risk.
 *
 * @author Michael Horn
 * @version $Revision$, $Date$
 */
public class Viewer extends Applet implements
Runnable, MouseListener, MouseMotionListener, WindowListener {

    /** Dimensions of the window */
    public static final int WIDTH = 400;

    private ConvexHull hull;           // Implements the algorithm
    private Canvas3D canvas;           // Renders the 3D convex hull
    private Toolbar buttons;           // The control buttons
    private Image offscreen;           // for double buffering
    private Thread thread;              // controls spinning
    private boolean spinning;           // whether the hull is spinning
    private boolean mousedown;         // current mouse state
    private Matrix rotation;           // stores the user driven rotation
    private int width;
    private int height;

    /** *****
     /
    public Viewer() { }
```

```
/**
 * Applet has been loaded into the system.
 */
public void init() {
    this.hull = new ConvexHull();
    this.canvas = new Canvas3D();
    this.offscreen = null;
    this.spinning = true;
    this.mousedown = false;
    this.buttons = new Toolbar(this, 0, 0, getWidth(), 30);
    this.buttons.addButton("Restart");
    this.buttons.addButton("Step");
    this.buttons.addToggleButton("Spin");
    this.rotation = new Matrix();
    this.thread = new Thread(this);
    this.thread.setPriority(7);
    this.thread.start();
    this.width = getWidth();
    this.height = getHeight();

    setBackground(new Color(220, 220, 220));
    addMouseListener(this);
    addMouseMotionListener(this);
}

/**
 * Callback for button clicks
 */
public void action(String command) {
    if (command.equals("Step")) {
        this.hull.step();
    }
    else if (command.equals("Restart")) {
        this.hull.restart();
    }
    else if (command.equals("Spin")) {
        this.spinning = !this.spinning;
    }
}

/**
 * Paint simply calls update
 */
public void paint(Graphics g) { update(g); }

/**
 * Draws the convex hull and buttons to the screen.
 */
public void update(Graphics graphics) {
    if (offscreen == null ||
        width != getWidth() ||
        height != getHeight()) {
        offscreen = createImage(getWidth(), getHeight());
        buttons.setWidth(getWidth());
        width = getWidth();
    }
}
```

```
        height = getHeight();
    }

    Graphics g = offscreen.getGraphics();
    g.clearRect(0, 0, getWidth(), getHeight());

    float w = (width < height)? width : height;
    float scale = 0.65F * w / hull.getDiameter();
    Matrix m = this.hull.getMatrix();
    m.identity();
    m.mult(rotation);
    m.scale(scale, -scale, scale);
    m.translate(getWidth() / 2.0F, getHeight() / 2.0F, 0);
    this.canvas.render(hull, g);
    this.buttons.draw(g);

    graphics.drawImage(offscreen, 0, 0, null);
}

/**
 * Spin the convex hull...
 */
public void run() {
    while (true) {
        if (spinning && !mousedown) {
            this.rotation.rotate(0F, -2.0F, 0F);
            repaint();
        }
        try { Thread.sleep(50); }
        catch (InterruptedException ix) { ; }
    }
}

int prevx;
int prevy;

public void mousePressed(MouseEvent e) {
    int x = e.getX();
    int y = e.getY();
    if (buttons.mouseDown(x, y)) {
        repaint();
    }
    else if (!buttons.intersects(e.getX(), e.getY())) {
        prevx = x;
        prevy = y;
        mousedown = true;
        this.canvas.setPickPoint(x, y);
        repaint();
    }
}

public void mouseReleased(MouseEvent e) {
    mousedown = false;
    this.canvas.setPickPoint(-1, -1);
}
```



```
        if (buttons.mouseUp(e.getX(), e.getY())) {
            repaint();
        }
    }

    public void mouseMoved(MouseEvent e) {
        if (buttons.mouseMove(e.getX(), e.getY())) {
            repaint();
        }
    }

    public void mouseDragged(MouseEvent e) {
        int x = e.getX();
        int y = e.getY();
        if (mousedown) {
            float xtheta = ((float)(y - prevy) * 360F) / getWidth();
            float ytheta = ((float)(prevx - x) * 360F) / getHeight();

            this.canvas.setPickPoint(x, y);
            this.rotation.rotate(xtheta, ytheta, 0F);
            repaint();

            prevx = x;
            prevy = y;
        }
    }

    public void mouseClicked(MouseEvent e) { }
    public void mouseEntered(MouseEvent e) { }
    public void mouseExited(MouseEvent e) { }
    public void windowActivated(WindowEvent e) { }
    public void windowClosed(WindowEvent e) { }
    public void windowClosing(WindowEvent e) { System.exit(0); }
    public void windowDeactivated(WindowEvent e) { }
    public void windowDeiconified(WindowEvent e) { }
    public void windowIconified(WindowEvent e) { }
    public void windowOpened(WindowEvent e) { }

    public String getAppletInfo() {
        return "3D Convex Hull Visalization";
    }

    public static void main(String args[]) {
        Frame frame = new Frame("ConvexHull");
        frame.setSize(WIDTH, WIDTH);
        Viewer v = new Viewer();
        v.setSize(WIDTH, WIDTH);
        v.init();
        frame.add(v);
        frame.addWindowListener(v);
        frame.setVisible(true);
    }
}
```